Order processing using a computer

ORDER PROCESSING USING A COMPUTER

by

STEPHEN GEOFFREY MINSHULL

A Master's by Project submitted in partial fulfilment of the requirements for the award of the degree of MSc in Computer Education of Loughborough University of Technology, January 1988

Supervisor: DR. R.G.STONE, BSc, PhD, MBCS, MACM.

ABSTRACT

This project report describes a package designed to teach business studies students some of the principles of information processing. It is particularly relevant for students following BTEC courses (the Business and Technician Education Council). The package simulates the processing of orders in a small firm. The students' tasks are to maintain master files, enter orders, print invoices, and produce and interpret reports. There are help screens available throughout the package which explain various aspects of information processing, as well as giving some guidance on using this particular package.

Students who use the package should obtain a good understanding of concepts which they will meet initially in the classroom and later when they use computers in their work.

The package fills a gap identified by the author in software available - commercial software being too complex, and educational software being too simple for the students at whom this package is aimed.

The package runs on an IBM PC or compatible, using dBASEIII+ as the programming language.

CONTENTS

APPENDICES

Section 1 - Aims of the Package

This Order Processing Package is designed to help with the teaching of a number of information technology modules which are found on business studies courses, especially in Colleges of Further Education. In particular, it has been written for the Information Processing 1 module of the BTEC Certificate/Diploma in Business Studies. However, it would also be useful for courses such as the BTEC Certificate/Diploma in Computer Studies, which has a similar Information Processing module. The package could also be used in other courses which require students to have some knowledge of information processing, without requiring a detailed knowledge of computers. However, this section will look at the aims of the program in the context of teaching IP1 to a BTEC Diploma in Business Studies group (which is also the group used for the pilot testing).

It is necessary first to examine the learning objectives of the IP1 module. The module is:

"intended for those students who wish to acquire skills in the techniques of using micro-electronic technology to provide information for the control and operation of a business. It emphasises the practical uses to which the technology can be applied and describes the constraints and benefits of using computers in the business environment." (J185, 1983)

The BTEC aims are:

- to develop an understanding of the role and range of technology available for information processing,
- to develop an understanding of the techniques available to provide information for the control and operation of an

organisation,

- to be able to use the technology to solve business problems.

The above aims are broken down into 8 learning objectives, which MUST be covered for satisfactory completion of the module (each objective being assessed by means of an assignment):

A) discuss problems related to business, and identify how information technology systems could be used to help solve these problems,

B) be aware of the types of information essential to the operation and control of a business,

C) understand the basic principles of collection, validation and processing of information,

D) be aware of the different types of hardware available for information processing,

E) be aware of the work of data processing specialists in devising cost effective solutions to business problems,

F) identify the role of the electronic office in information processing in business,

G) be aware of the practical uses of computers in business,

H) using available technology choose an applications package and use this package to operate a small business system.

The Order Processing Package is designed to be useful for a number of these objectives, in particular Objective H; however, it will also contribute substantially to meeting objectives B,C and G.

Section 2 - Brief Description

The package simulates the operation in a small business of the processing of orders, printing of invoices and production of reports. It has been designed to be as simple to use as possible. Throughout the package, selection of options is by moving a highlighted bar to the correct option using the cursor keys, and selecting that option by pressing return. A status bar at the bottom of the screen will always indicate if there is a help screen available; the current part of the package; and the date. The help screens are available throughout the package, and are used to explain different aspects of information processing on a computer. For example, in the backup section of the package the help screens will explain why backups are necessary; in the order entry section, the help will explain what batch processing is, and its advantages and disadvantages. There are over 20 different screens available.

There is no student user documentation, except that required to actually enter the package; however, it is recommended that the lecturer first explains the nature of the package, the situation which it is intended to simulate, and gives a brief demonstration. The student WILL require documentation concerning the exercises to be carried out, and a sample of these - the exercises used in the piloting - is given elsewhere in this report.

The features in the package include:

- maintenance of Customer, Product and Supplier Master files;
- orders entered singly or in batch mode;
- processing of payments to suppliers, and from customers;

- extensive reporting, including sales ledger, audit trail, printing of address labels, statements, bar graphs, etc., as well as the more usual reports expected in such a system;
- backing up and restoration of data files;
- a large number of help screens;
- entry and maintenance of passwords, allowing different levels of access;
- changing the default colours and name of the company.

If the full version of dBASEIII+ is being used, file sizes are limited only, in practice, by disk capacity. However, if the sampler version of dBASEIII+ is being used, file sizes are limited to 31 records. The implications of this are discussed in more detail in the section giving instructions to lecturers.

Section 3 - Review of Existing Software

This can be split into two distinct types - software written for commercial use, and adapted for education (eg. the AXIS and Sage accounting packages); and software written for educational use (eg. the Pitmansoft and Acornsoft ranges).

a) Commercial software

There are various problems related to using commercial applications packages in an educational environment:

i) In general, commercial software is too complex for educational use (although this does not apply to generic packages such as spreadsheets, where the lecturer can set up relatively simple exercises using only the commands required for that exercise). For example, the AXIS Modular Accounts Package offers a number of modules, including stock control, purchase and sales ledger, invoicing, etc. However, the initial setting up of the company is very complex, requiring the entry of a variety of different data before any useful exercises can be developed. This is, of course, what happens when a 'real' company moves from a manual to a computerised accounting system - the actual changeover period, the setting up of the computer system, is very time consuming.

ii) Following on from the above, the lecturer will often find that the package offers too many facilities - facilities which may be needed for commercial use, but which simply confuse the student with a lot of unnecessary information on the screen. In the great majority of cases, the students will not have the detailed knowledge of book-keeping/accounts required to understand what the package is actually doing. This will detract

from the main aim of the work, which is to understand how computers can be used in business - NOT to teach students how to do accounts on a computer, or to get to grips with using a particular package.

iii) Help screens and tutorials - if available on the package - are aimed at teaching users how to operate that particular package, rather than giving an understanding of the principles underlying information processing.

iv) The lecturer will normally have a series of exercises which need to be carried out using the package. The same exercises will need to be carried out by a number of different classes over the year. This means that each time a new class uses the package, the data needs to be 'reset' to the original. Although always possible, it can be a lengthy and complex business, particularly for a lecturer who is not a computer specialist.

v) Commercial packages tend to be more expensive than educational software. The educational institution will inevitably require multiple copies of the package, and unless the supplier offers very favourable terms, the cost of purchase can be considerable. The educational price for 10 copies of 'Accountant Plus' from Sagesoft is 875.00 - cheap in commercial terms, but a considerable hole in the budget of most colleges. (The Order Processing Package is written in dBASEIII+, and can be used with the sampler version of dBASEIII+, which is supplied virtually free to educational institutions. Apart from the file size restriction mentioned above, the sampler version has all the facilities of the full version.)

b) Educational software

The information for this section was obtained from the Further Education Unit Courseware Directory, 1987 edition.

i) The vast majority of educational software of this type is written for the BBC B or Master computers. These use a non-standard operating system, and are unlikely to be found outside education or the home. The trend in F.E. colleges is to use typical small business micros, such as the IBM PC. There is, at present, very little educational software written for IBM compatibles.

ii) Much of the software that is available on the BBC tends to be too simple to meet the requirements of courses such as BTEC Business Studies. This is partly a reflection of the fact that the software was written to fit the small size of the BBC, and this still applies, even though expanded models are available. The FEU Courseware Directory lists only 3 packages which are similar in objectives to the Order Processing Package.

'Micros in Business', Acornsoft. 1985

This package includes a spreadsheet, a word processor, a database and a planner. It takes students through a series of demonstrations where the student works through role-plays of tasks which might occur in the running of a small firm. The package is aimed at pre-vocational business studies students, and whilst adequate for this task, is too simple to meet the needs of a higher level information processing course. Only available on the BBC.

'Stock', System Software, 1984

This is very similar in outline to the Order Processing Package, but is far more limited in its options - for example, it does not produce a stock valuation report. It is really aimed at less experienced students - eg. those on BTEC First Diploma and Certificate courses, which also have an information processing content. Avalailable on RML 380Z, AppleII, and BBC B.

'Accounts II', Five Ways Software and MEP, 1984

This is aimed at Business Studies students who are taking accounting/bookkeeping, and goes into accounts in too much detail for the purposes of Information Processing options. Students are required to draw up trial balances, do double entry book-keeping, etc. Available only on the BBC B.

There is other educational software available, but it is almost invariably for the BBC; and is either too simple, eg. offering inadequate reporting; or too narrow in scope, eg. designed to teach accounts.

In conclusion, therefore, as far as I am aware, there is no software available which teaches the principles of information processing at the BTEC National level (equivalent to 2 'A' Levels); which is on IBM compatible machines; and which offers such extensive help and reporting features.

Section 4 - The Programming Language

The language chosen was dBASEIII+. This was partly because a demonstration version of the program is available at little cost to educational institutions, although file size is restricted to 31 records; and partly because of the nature of the language itself.

dBASEIII+ is a high level relational database, with its own programming language, the successor to dBASEII. Like dBASEII, it facilitates structured programming with DO CASE and DO WHILE statements, and separate program modules can be split up into separate procedures. However, there are various improvements over dBASEII. These include:

- 10 data files open at any one time (2 in dBASEII)
- more functions available, eg. INKEY, BOF(beginning of file)
- explicit declaration of variables as PUBLIC or PRIVATE (to a procedure)
- more PICTURE options to allow formatting of input and output
- the ability to have a procedure file held in main memory when not being executed, so that it does not require a disk access when called (the help file is accessed in this way)
- a pseudo-compiler which tokenises key words and strips out spaces and comments, and considerably decreases execution time.

Section 5 - Overview of System

i) Input:

- maintenance of master files (Customer, Product, Supplier)
- orders from customers
- payments from customers
- payments to suppliers.

ii) Processing:

- process orders and payments to update master files
- process data for invoices
- process data for other reports.

iii) Output:

- invoices, statements, ledgers, and other reports.

Section 6 - Data Structures

The detailed file structures and associated indexes appear in the appendix.

i) The CUSTOMER and PRODUCT files store codes, names and addresses, as well as information on the balance and credit limits, and total sales or purchases to date.

ii) The PRODUCT file stores product codes, product group, supplier code, prices, quantity in stock, minimum and maximum stock levels, and reorder quantities.

iii) The PASS file stores user IDs and passwords, and their access level

iv) Orders are stored in the ORDER file, which consists of the customer code, product code, quantity and date. There are also various booleans:

UPDATED is true after an order has been processed and the master files updated - so an order cannot be processed twice, or deleted after being processed. If there is insufficient stock to meet the order, UPDATED remains false. The file SCRATCH is used to temporarily store these records, which are printed at the end of the updating routine.

INVOICED is true after an invoice has been printed, so that an invoice cannot be produced twice. (At the same time, the invoice number, date and total are written to the file for use later when statements are produced)

LEDGERED is true after the sales ledger has been printed.

TRAILED is true after the audit trail has been produced. At this point the user is given the option of deleting old records - this is only done if all the above booleans are true.

v) PAYMENT records the customer payments made in response to invoices, and is needed for the production of statements.

vi) PRODHIST is used in the production of barcharts, and records are written during the invoicing printing routine. If this extra file were not used, it would result in a considerable increase in processing time during bar chart creation.

vii) CUSTBACK, PRODBACK, SUPPBACK and ORDBACK are produced when the main data files are backed up - ie. CUSTOMER, PRODUCT, SUPPLIER and ORDER.

viii) SCRATCH has 5 fields, F1 to F5, and is a temporary storage file - hence the field names.

Section 7 - The Program Logic

The program was written in modules (equivalent to PRG files) which were individually tested. In the final version, these modules have been compiled and linked together into one file. The listings are still in the modular form, although certain PRG files have been absorbed into others, eg. the invoice routine was written and tested as a separate module, but is now incorporated into ORDER.PRG. The reason for this is that dBASE can only have one routine, apart from that being executed, resident in main memory. This is used for the help routine, HELP1.PRG. Therefore every time a PRG file is called, a disk access is required, which noticeably slows the program. Therefore, where convenient, the number of PRG files has been reduced. However, the larger the file, the longer the execute time - this is particularly noticeable during option selection. Therefore not all files have been combined into one.

A problem arises with the number of files which can be open at any one time. This is 15 in total - ie. including data files, index files, procedure files, and program files. One of the more extensive routines may require, for example, the Customer, Product and Order files, PLUS their associated indexes PLUS the HELP1 file which is resident in memory, PLUS up to 3 program files if the routine is at the bottom of the hierarchy. The total can very quickly reach 15+, causing the package to crash. Hence a number of different levels have been incorporated into one, even though developed individually - thus the INIT routine now also contains the password routine and the main menu routine.

The rest of this section looks at the various program modules,

and should be read in conjunction with the structure diagram and the listings. All files (in **UPPER CASE**) are assumed to have the PRG extension. If the data disk you are using has had the files LINKed together, there will only be 2 PRG files - HELP1 the help file; and INIT, the main file. Otherwise all the PRG files will be on your disk. The files are discussed from the top level downwards, with the exception of HELP1, EHANDLER and DRIVE, which are discussed first, and ENTORD, which is discussed with ORDER. Password access levels other than 1 are given in brackets.

DRIVE

This routine is run when using the package for the first time, or with a new program disk. It is called by invoking at the system prompt:

```
dbase n:drive
```

where n is replaced by the drive with your data disk. It stores the data drive letter onto the PROGRAM disk, in the file DRIVE.MEM, so that in future the package is started by invoking at the system prompt:

```
dbase n:init
```

This makes the package reasonably flexible with respect to hardware.

EHANDLER

This is the error handling routine, called up whenever dBASEIII+ encounters an error. This is most likely to occur if a file exceeds 31 records when using the sampler version of dBASEIII+ (see Appendix A). If an error does occur, the error message is printed on the status bar, and suitable action taken by the routine.

HELP1

This routine resides in main memory, and has all the help screens split into different named procedures - the relevant procedure is called by pressing the F1 key during the program.

INIT

This routine sets up various system parameters, eg. the keyboard buffer; recovers the company name and screen colours from MEM files; sets the date to the British format; etc. Files are opened and work areas selected. The initial help screens are shown, and the password entry routine begins. If the password is entered successfully, INIT continues, otherwise, after 3 attempts, the package quits to the system prompt.

Assuming INIT continues, it goes into the main menu routine, with 7 options:

i) Master files. This simply calls the relevant PRG files.

ii) Orders. Calls ORDER.

iii) Payments. Calls SPAY and CPAY

iv) Reports. Calls REPORTS.

v) Backup. Calls BACKUP.

vi) Installation. Calls ENTPASS to maintain password file, and COLS to change colours. Allows company details to be changed, and stored in COMPANY.MEM. Allows user with highest level of access to zap old files (eg. as the lecturer would have to do to do 'reset' the data for a fresh class).

vii) Exit. Returns to the operating system.

CUSTOMER, PRODUCT, SUPPLIER

These routines are essentially the same. They each allow the user to skip backwards and forwards through the file; to find by

shortname (see the section on the user interace for an explanation of this); to add, change and delete records.

ORDER

This calls ENTORD to actually enter orders.

The 'Process Orders' option updates the Product File, reducing the quantity in stock by the amount of the order. If there is insufficient stock, the record is temporarily stored in SCRATCH.DBF, and printed out at the end of this routine. The order record is marked as having been processed.

The invoices option sends invoices to screen or printer. They can only be printed once, when the customer file is updated and the product history file created. The invoice details are written to the order file.

ENTORD

This has 5 options (plus an exit option):

i) Singly. Orders are entered one at a time, ie. there are NO batching or hash totals. The user enters an order number; the short name of the customer, after which more customer details are printed, and the user has to confirm the customer - if not the correct customer, the user can skip to the next customer, alphabetically, or exit; the product, which is checked in the same way (only the first few letters need be entered); and the quantity. The order is then written to the order file.

ii) Batch. The routine is essentially the same as above, except that first the user has to enter the total number of items in the batch; and the total of the quantities. If the totals at the end of the batch entry do not match, the batch must be re-entered, otherwise it is written to the order file as above. A batch can

have up to 99 items (lines on orders) but it is recommended that batches are small, 10 maximum, since students will probably enter them incorrectly at first, and so have to re-type that batch.

iii) Delete. Orders can be deleted, but ONLY if they have not been processed. Orders are found by order number (unlike the delete routines in the master files, where the records are skipped through using the NEXT and PREVIOUS options. This is to illustrate 2 different ways of searching a file.)

iv) Change. The order quantity can be changed, but again only if they have not been processed.

v) List. List orders to screen or printer.

CPAY

This processes customer payments. The customer is located using the search on the short name, with confirmation by the user, and the payment entered. The balance field in the Customer File is updated, and the payment details are written to the PAYMENT.DBF file for use in statement production.

SPAY

This is similar to CPAY, processing payments to suppliers. The Supplier File is updated, but since payments' statements are not produced, the data is not saved to any other file.

BACKUP (3)

This allows the backup or restoration of customer, product, supplier and order data, having first checked that there is sufficient diskspace to do this. It only backs up to the same disk. On twin floppy machines, this is the only way to do it, UNLESS the backup is done via the operating system, which is not really feasible for this exercise. Backups could easily be made

from floppy to hard disk, but it was considered that backing up to the same floppy was enough to illustrate the point of the importance of making backups.

On restoring from backup, the data files are re-indexed.

ENTPASS (4)

This allows the user to enter new users, passwords and access levels, as well as changing and deleting existing users.

COLS

This allows the user to change the text foreground and background colours; and the helpscreen foreground and background colours. The user is not allowed to make foreground and background the same colour.

REPORTS (2)

This has 14 options (plus exit). The menu is vertical rather than horizontal because of the number of options. These are as follows:

i) Lists the customers over the credit limit to screen or printer.

ii) Produces a standard letter to be sent to customers over the credit limit.

iii) Produces statements. Calls STAT.

iv) Lists all the customers to the printer

v) Produces a product re-order list. Calls REOMAX.

vi) Prints a standard order to be sent to suppliers when products need re-ordering.

vii) Prints out a stock valuation by group.

viii) Lists stock over the maximum to screen or printer.

ix) Prints a list of all the products.

x) Produces bar charts. Calls BARCHART.

xi) Lists all the suppliers to the printer.

xii) Produces a sales ledger. Calls LEDGER.

xiii) Prints an audit trail. Calls AUDIT.

xiv) Prints address labels. Calls MAILLIST.

STAT

Produces customer statements. This uses the Customer, Payment and Order Files. It reads through the Order File for records which have the boolean 'INVOICED' set; and prints all the records for the same customer (hence the Customer File is used to get the correct customer details). The Payment File is then checked to see if there are any payments from that customer. If so, they are printed. The statement balance is then printed, and the 'STATEMENT' boolean set in the Order File.

The routine then returns to the Order File to find the next order(s) for the next customer.

REOMAX

This routine prints reports for stock under minimum, and stock over maximum - this is controlled by the variable 'max' which is set by the calling routine.

BARCHART

This prints to the screen two barcharts - stock valuation by product group at cost price; or sales by product group at sale price. The latter option requires the use of the PRODHIST.DBF file. The graph is automatically scaled.

LEDGER

This produces a simple sales ledger using the Customer and Order Files. The ledger can be sent to screen or printer - after

printing, the boolean 'LEDGERED' is set in the Order File.

AUDIT

This produces 2 audit trails - one for payments, and one for ledgers. After the trail is printed, the user can delete old records - IF they have been PROCESSED, INVOICED and LEDGERED first. After deletion, the entry will no longer appear on customer statements, whether or not it has actually been paid. In this educational package, this should not be a problem.

MAILLIST

This allows the automatic production of address labels. Having selected the file to use (Supplier or Customer) the operator then has the choice of selecting all the addresses, selecting by name, or by town, or where balance is greater than credit. This module uses a dBASEIII+ address label routine, which unfortunately can only handle one file at once - it is therefore not possible, for example, to print out all addresses where stock needs re-ordering, because that involves 2 files. Obviously, this could have been done by creating an additional file just for mailing; but this was not considered necessary for this package.

Section 8 - The User Interface

i) Menu selection is usually achieved by moving a highlight across the menu options, and pressing RETURN to select. As the highlight moves across, a brief description of each option appears on the line immediately below the options. (See fig. 1 - note that the screen dumps cannot show inverse video, ie. the highlight, which affects the status bar and menu options; and that boxes drawn on screen appear as columns of '3's, and rows of 'D's).

The menus are as standardised as possible - eg. the options on the Customer, Product and Supplier Files are identical. All menus are horizontal, except the reports menu where the number of options necessitated a vertical menu (fig. 2) The HOME and END keys will go to the first and last option respectively. All other keys, except RETURN, the F1 help key and the relevant cursor keys, are inoperative.

Selection by initial letter is not implemented, mainly because (except on the reports module) the list of options is so short as to render this unnecessary. This method of selection was chosen partly because of its simplicity, but also because it is becoming increasingly common.

If there is a mouse attached to the computer, this can be used for moving the highlight and selecting an option.

In certain cases selection is by entering a letter and pressing RETURN (fig.3). This is usually done at the bottom level of the menu hierarchy, where options are very limited (eg. to send to screen or printer) Using the highlight method at this level would have resulted in a considerable increase in programming for a

Jane's Boutique Ltd.

Master files Orders Payments Reports Backup Installation Exit
Use customer, product and supplier files

<F1> for help main menu 06/11/87

Figure 1

Jane's Boutique Ltd.

Customers: list of those over credit limit
Customers: letter to those over credit limit
Customers: statements
Customers: list all customers to printer

Products: reorder list
Products: order to supplier
Products: stock valuation by group to printer
Products: stock over maximum list
Products: list all products to printer
Products: produce bar charts

Suppliers: list all suppliers to printer

Sales ledger
Audit trail to printer
Address labels

Return to previous menu

<F1> for help reports menu 06/11/87

21b

Figure 2

You can get a Screen report - S Enter your choice:
or a Printed report - P
or eXit - X

Figure 3

fairly trivial return.

Using pull down menus was considered, but again the increase in programming, and consequent slow down in execution time, was considered too great.

The keyboard buffer has been reduced to one keypress. This is to avoid the situation of students holding down keys, and causing the highlight to cycle continuously through the options. In certain places the buffer is cleared, to avoid falling through from one menu to another because of the stored keypress.

The package was written on an Amstrad 1512, and there were no delays in pressing a cursor key, and the highlight moving. On slower machines, such as the IBM PC, there may be a slight delay on the first keypress when entering a different menu.

ii) In the master files, a record is located by entering the SHORTNAME. If part of the shortname is entered, the first record containing those letters will be located, and then the user can move to the next one if this is incorrect. The order entry routine works in a similar way, where the user enters all or part of the customer SHORTNAME, but this time s/he has to confirm that it is the correct customer; then enters all or part of the product description, and again has to confirm or reject the product (fig. 4). The user then will enter the quantity. To move back to the customer (eg. to start a new order) the user presses return instead of entering anything when asked for the product. To finish entering the batch, the user presses return when asked for the customer.

iii) The screen layout is also designed to be consistent, with a status bar at the bottom which indicates which part of the

Jane's Boutique Ltd.

Singly Batch Delete Change List Exit

Enter orders one at a time

```
ZDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD?
3 Order num:    103                        3
3 Short name: CRISSY                       3
3 Full name:  CRISSY FOODS LTD             3
3 Address:    45 NORTH RD                  3
3                                          3
3 Description:  JORDANS CRUNCHY BARS       3
3                                          3
3                                          3
3                                          3
3                                          3
3                                          3
3 Correct product? Y/N or X to quit:       3
@DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDY
```

process one at a time 06/11/87

22a

Figure 4

program the student is in, as well as the date and help indicator. In general, data entry masks are on the left of the screen; help screens on the right (fig. 5).

iv) When available - which is in the great majority of options - the help screens are called up by the F1 key. If there is more than one frame (the longest is 4 frames) the student progresses by hitting any key. The cycle only works one way.

v) The user can install the colour to suit his or her needs. Text foreground and background, and helpscreen foreground and background, are user definable. The program does not allow the user to have foreground and background in the same colour. If the colour options are used on a mono monitor, they appear as various shades of grey.

vi) The user is always able to exit from an option, so long as data has not been entered. Exit is by pressing return in the first field. The ESCAPE key is not implemented.

vii) dBASEIII+ has a number of editing keys available during data entry - eg. CTRL T will delete the rest of the line. These have NOT been described in the student documentation, since my experience has shown that initially students prefer to use the cursor control keys, the delete key, and the insert key to edit lines. Listing all the other editing keys adds unnecessarily to the complexity, and detracts from the real aim of the program.

viii) The cursor has not been switched off. Ideally, it should be switched off during the menu selections, and switched on during data entry. However, this would have meant loading in another 2 binary files to do this, since there is no routine within dBASEIII+ to control the cursor. However, as previously

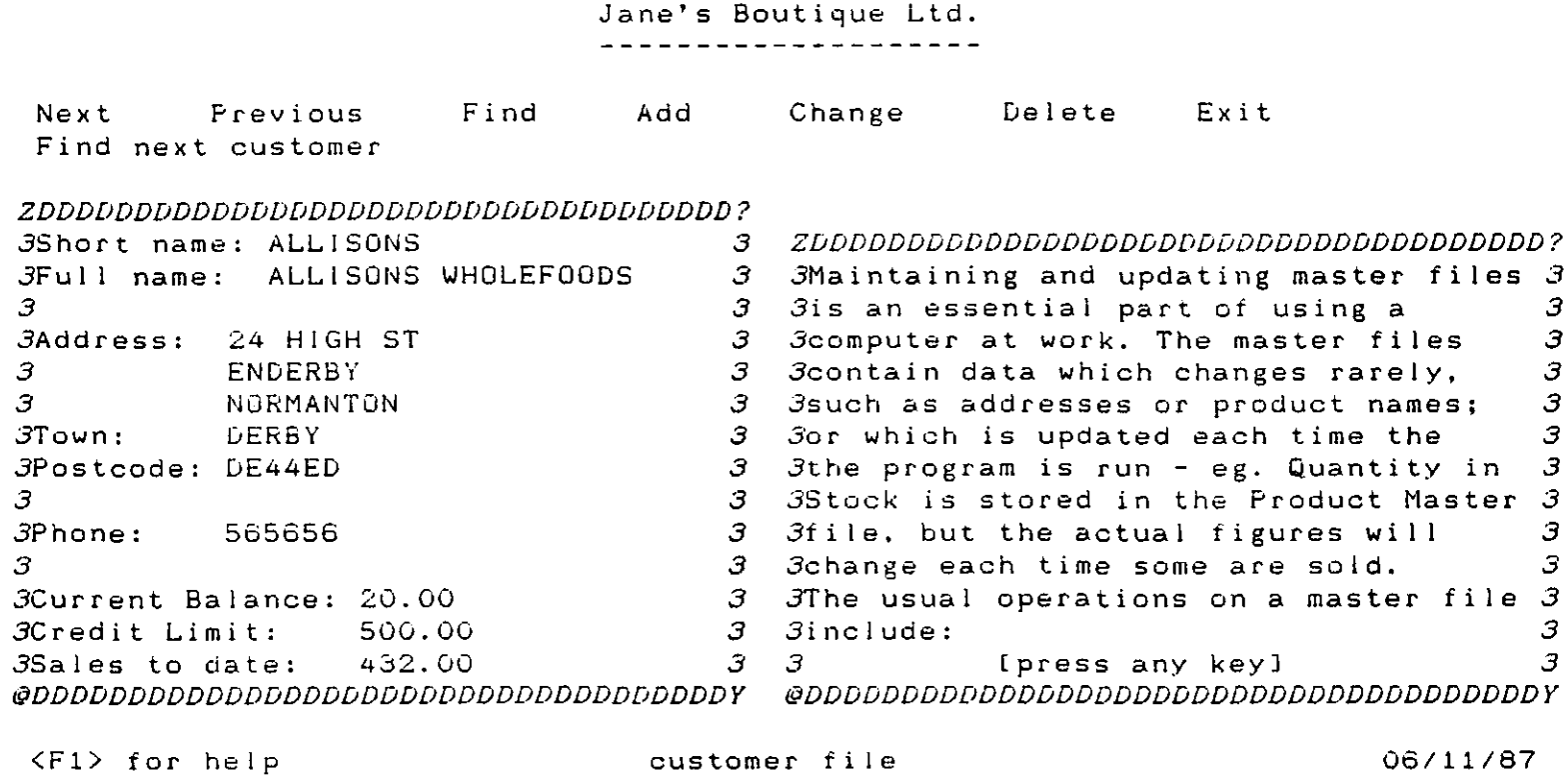

Jane's Boutique Ltd.
Next Previous Find Add Change Delete Exit
Find next customer
Short name: ALLISONS
Full name: ALLISONS WHOLEFOODS
Address: 24 HIGH ST
ENDERBY
NORMANTON
Town: DERBY
Postcode: DE44ED
Phone: 565656
Current Balance: 20.00
Credit Limit: 500.00
Sales to date: 432.00
Maintaining and updating master files is an essential part of using a computer at work. The master files contain data which changes rarely, such as addresses or product names; or which is updated each time the the program is run - eg. Quantity in Stock is stored in the Product Master file, but the actual figures will change each time some are sold. The usual operations on a master file include:
[press any key]
<F1> for help
customer file
06/11/87

Figure 5

mentioned, only 15 files can be open at any one time. Hence having 2 additional files would have required a substantial amount of reprogramming. A decision on whether or not to do this was deferred until after the testing. In fact, the students using the package were not confused at all by the cursor, and therefore it has been left in the package.

PLEASE NOTE:

Every effort has been made to remove any bugs. Nevertheless, without more extensive testing - including the use of the package by lecturers with little or no computing knowledge - it would be foolish to claim that it is bug-free. However, the error handling routine should ensure that any problems which do arise (except hardware problems, corrupted disks, etc.) should NOT cause the package to crash, although an error message will appear on the status line, following which processing will continue.

Section 9 - Test Results

The package was tested over a 2 week period at the beginning of October 1987 with a class of 20 students doing the first year of a BTEC Diploma in Business Studies course at Derby College of Further Education. The students had only completed 4 weeks of information processing, and their knowledge was minimal. Their practical experience to date was limited to 3 lessons on Wordstar.

Ideally, on this particular course, the package should be used in the Spring term, when the students are more familiar with information processing, and when they have to begin a major project linking together the different objectives of the Information Processing 1 course (see Section 1). However, because of the deadline for submission, it had to be tested at this relatively early date. This meant that the students were being introduced to concepts - such as batch processing - without having discussed them in theory classes first.

The testing had 2 aims:

a) to see how the package stood up to use in a classroom situation

b) to attempt to evaluate its educational value.

a) Classroom use

The package was used by 20 students working in groups of 2, each group with its own 20Mb hard disk Amstrad 1512. All files - program and data - were stored in a subdirectory of the hard disk. Printers were shared between 2 or 3 computers - when someone wanted to print they had to switch the printer sharer unit to their machine. Each group of students worked through the

exercise 'Using a Computer For Your Business' (Appendix B). The function of the package was briefly explained, but the software itself was not demonstrated.

Results:

i) The biggest problem was with printing. If a printer was switched off, or not connected to the computer, MS-DOS threw up an error message onto the screen which was not trapped by the error handler in the package. Obviously, this affects the screen display, until the program re-draws the screen. (If the printer was off-line, or switched to another computer, there was no problem - the computer simply hung until it was remedied). I subsequently spoke to the Ashton-Tate technical support staff, who informed me that it is NOT possible to trap this type of error from within dBASEIII+. The only solution is to make sure that the printer is actually switched on before attempting to print. In fact, a screen message to this effect appears before every print operation.

ii) The F1 help key - which unlike the other function keys cannot be re-programmed to nul - terminated data entry if pressed during a READ operation. This has now been corrected.

iii) One of the help screens was drawn in the wrong place on the screen, and overwrote other text. This has been corrected.

iv) The routine which allows the user to change colours presented a slight problem. Although the text on the screen told the students not to make the text the same colour as the backgound, some did - with the obvious result that they could no longer read any instructions. The COLS.PRG program has now been amended so that it is not possible to have background and

foreground the same colour.

iv) With the exception of the points above, the package held up very well - and in each of the cases above the package did not crash, and the students were able to continue their work with only a limited interruption.

v) The students found the package very easy to use - on completing the exercise sheet, they were encouraged to experiment with the various options, including those not covered in the exercise. They did this without any problems.

b. Educational value.

In the last lesson, the students had to answer a number of questions, which were designed to test how much they had absorbed from working with the package. The question sheet is in Appendix C.

i) Judging from their answers, the students all enjoyed working with the package. This was pleasing, because within groups like this, there are always some students who find computers 'boring' or who 'hate them' or who 'cannot understand any of it'. Motivating students with these attitudes can be difficult. The fact that they could relate the exercise to the world outside, and successfully carry out the tasks, is in itself important.

ii) The students learnt a lot about the practicalities of using a computer system at work - eg. the need for data backups. The full benefit of the package will come when used over a prolonged period to simulate, for example, a working month in the life of the company.

iii) The answers to the question sheet showed that they had assimilated a considerable amount of knowledge - eg. they now

have some idea of the use and role of the different reports in a system such as this; they are aware of the need for security; etc. However, the real benefit will come when these techniques (eg. batch and real time processing) are discussed in more detail in the theory lessons in class later on this term. They will, through the practical exercises, have gained a good grounding in some of the concepts and techniques which they will then meet in class.

iv) One problem which became apparent from the answers was that some students (in spite of the careful wording of the help screens) thought that the way something was done in this package was the ONLY way to do it. Eg. a number of students wrote that an audit trail only consisted of lists of orders and of payments - which is the trail produced by the package. A different application would, of course, have a different type of audit trail. However, this type of problem should be overcome by the classroom teaching which supports the package.

Brief analysis of Answers

Refer to Appendix C for the questions.

This is not intended to be a rigorous analysis, in any way. The question sheet was devised simply to give a rough guide to how much the students had assimilated whilst using the package. However, bearing in mind that these students had only just started the IP1 course, and therefore most of the concepts were quite new to them, the results were encouraging. 20 answer sheets were completed. The following figures give the percentage of correct answers to each question:

Q1. 85%

Q2. 95% - satisfactorily explained the relationship between files, records and fields.

Q3. 85% - although the detail of the replies varied considerably here.

Q4. Only 20% managed to give a correct example - none was actually given in the help screens.

Q5. 95%

Q6. 90% gave a partially correct answer, limiting their example to deletion of files.

Q7. 90% answered this correctly, although their answers were limited to the example of the package.

Q8. 85%

Q9. 90%

Q10. Without exception, all students found the package easy to use, and enjoyed the work. The only part which some found difficult initially was the batch entry - but after 1 or 2 attempts, all students understood how to do it, and completed the tasks successfully.

CONCLUSION

Considering that the package had to be tested earlier than desirable on such a course, response from students was favourable. But possibly equally important, is that a number of non-computing staff from the Business Studies Department who are involved in courses with an information technology element have examined the package and believe that they would find it easy to use on their courses, and beneficial to the students involved. This is, of course, very important. There is little point in

developing a package unless it is going to be used very widely, including use by non-technical staff. However, from comments received during the testing, it is apparent that the package WILL be used by other staff who are not familiar with computers.

Section 10 - References

1. BUSINESS AND TECHNICIAN EDUCATION COUNCIL (1983): Information Processing 1 (J185)

2. FEU COURSEWARE UNIT(1987): Courseware Directory.

3. Micros in Business (1985). Published by Acornsoft Ltd.

4. CLARKE, J. (1984): Stock, published by Arnold Wheaton for System Software.

5. LINEY, B. and PATERSON, J. (1984): Accounts II, published by Heinemann Computers in Education Ltd. for Five Ways Software and MEP

Appendix A - A Guide for Lecturers

This Order Processing Package is designed to help with the teaching of a number of information technology modules which are found on Business Studies courses, especially in Colleges of Further Education. In particular, it has been written for the Information Processing 1 module of the BTEC Certificate/Diploma in Business Studies. However, it would also be useful for courses such as the BTEC Certificate/Diploma in Computer Studies, which has a similar Information Processing module. The package could also be used in other courses which require students to have some knowledge of information processing, without requiring a detailed knowledge of computers.

The package simulates the operation in a small business of the processing of orders, printing of invoices and production of reports. It has been designed to be as simple to use as possible. Throughout the package, selection of options is by moving a highlighted bar to the correct option using the cursor keys, and selecting that option by pressing return. A status bar at the bottom of the screen will always indicate if there is a help screen available; the current part of the package; and the date. The help screens are available throughout the package, and are used to explain different aspects of information processing on a computer. For example, in the backup section of the package the help screens will explain why backups are necessary; in the order entry section, the help screen will explain what batch processing is, and its advantages and disadvantages. There are over 20 different screens available. However, please note that they are only intended as an INTRODUCTION to the various topics. It is

expected that, particularly for the higher level courses, the concepts introduced in the package will be backed up by additional work. For example, the explanation of the importance of audit trails would need to be expanded upon in class.

There is no student user documentation, except that required to actually enter the package; however, it is recommended that the lecturer first explains the nature of the package, the situation which it is intended to simulate, and gives a brief demonstration. The student WILL require documentation concerning the exercises to be carried out, and an example of these is attached.

The features in the package include:

- entry and maintenance of passwords, allowing different levels of access;
- changing the default colours and name of the company;
- maintenance of Customer, Product and Supplier Master Files;
- orders entered singly or in batch mode;
- processing of payments to suppliers, and from customers;
- extensive reporting, including sales ledger, audit trail, printing of address labels, statements, bar graphs, etc. as well as the more usual reports expected in such a system;
- backing up and restoration of data files.

The records maintained on the customers and suppliers are more limited than they would be in a 'real' situation. This is to avoid the package becoming too complex and confusing for the student. The package is teaching how computers are used in business - NOT how to do accounts.

A. REQUIREMENTS

The package is written in dBASEIII+, and therefore requires that software to run it. It will also run under the dBASEIII+ sampler version, which is available at nominal cost (approx. 1.00) from the publishers, Ashton-Tate. However, under the sampler each file can only have a maximum of 31 records. It is therefore ESSENTIAL that exercises using the sampler version are tested first, to ensure that files do not get too large; the lecturer should also clear out the old records whenever given the option during the running of the package, eg. after production of the audit trail. If the full version of dBASEIII+ is used, file size is no problem, restricted only by disk capacity.

Any IBM compatible computer using the MS-DOS or PC-DOS operating system which can run dBASEIII+ will run the package. A colour monitor is better than mono, but not essential. The package can use either a twin floppy, or a hard disk machine.

An 80 column printer is essential - preferably a dot matrix.

B. BACKUP THE DATA DISKS!

It is essential that you make a backup of the 2 data disks supplied. On a twin floppy machine, put the data disk in a: and the backup in b: Enter at the MS-DOS prompt:

copy *.* b:

Keep the original disks safe, and use the backups.

On a hard disk system, copy the files to the hard disk as detailed in the next section.

C. INSTALLATION

To run dBASEIII+, it is important that the config.sys file contains the line:

files=20

See your MS-DOS manual for information on this.

dBASEIII+ also allows you to set up some system parameters when calling dBASE, via a file called config.db. This file is NOT required for this package, since all the parameters are set up when the Order Processing Package is called. However, if you already have a config.db file set up, there should be no problem as long as it does not specify a default drive. If this is the case, delete that line from the file.

Possible configurations

i) Hard disk for Order Processing disk and dBASEIII+.

Ideally, you should already have installed dBASEIII+ in its own sub-directory, using the instructions in the dBASEIII+ and MS-DOS manuals. If the student data is going to be stored on the hard disk, rather than the floppy, copy all files from the OP disk into the dBASEIII+ sub-directory. Do this (making sure that you are in the dBASE sub-directory) by entering at the MS-DOS system prompt:

copy a:*.* c:

assuming that a: is the floppy drive, and c: the hard disk drive.

Then enter at the system prompt:

dbase drive

and follow the instructions.

ii) Hard disk for dBASEIII+, floppy for OP disk.

Put the OP disk in the drive, and key in at the system prompt:

dbase *n:drive*

and follow the instructions.

iii) Twin floppy machine

Use the left hand drive, a:, for the dBASEIII+ disk, and the right hand drive, b:, for the OP disk. Enter at the system prompt:

dbase *b:drive*

and follow the instructions.

D. SUBSEQUENT USE

At the MS-DOS system prompt, key in:

dbase n:init

where the drive reference n: is replaced by whatever drive you originally specified for the data.

E. SETTING UP USERS AND PASSWORDS

The password for staff use supplied with the disk is:

User id: STAFF

Password: STAFF

It is strongly recommended that you change this as soon as you get the package. Select the INSTALLATION option from the main menu (instructions on how to do this appear before you enter in the password); and then select ADD/DELETE USERS. When users are added, they are given access levels from 1 to 4 - this affects the routines within the package to which they have access, as follows:

ADD/DELETE USERS - level 4

BACKUP - level 3

REPORTS - level 2

All other routines are access level 1.

You should only have one user - yourself - with access level 4. It is suggested that you set up 2 or 3 other users with access

levels 1 to 3 to use with your classes.

Note that there is one other password included on the supplied disk - the password which is required for the sample student exercise sheet.

F. USING THE PACKAGE.

Two data disks are supplied with the package - the first one has some sample data on, and was the one used for the sample exercise included with the documentation. It is a good idea to work through these exercises, using this disk, before attempting to set up your own data.

The second disk is the one which you should use to set up your own firm. It has only one record in each of the master files. Put in your additional records before deleting these. Select the MASTER FILES option from the main menu, and enter the new data in each of the master files, using the ADD option. Delete the old data using the DELETE option, which deletes the record currently displayed on the screen. If using the dBASEIII+ sampler disk, the files must NOT exceed 31 records.

BACKUP the data disk so you can quickly restore the files - use the MS-DOS copy facility, rather than the backup routine within the order processing package.

Work out what exercises you want your students to do, and TEST THEM until you are sure they work.

When a new group starts the exercise, simply copy all the files over from the backup disk you made previously - which should, of course, be write protected.

G. ERROR-HANDLING

1. Printing.

If you attempt to print when the printer is switched off, or not connected to the computer, then MS-DOS will give an error message, together with the option to retry to print. If possible, connect the printer and retry. Unfortunately, the error message will spoil the screen display, until the whole screen is redrawn by the program. It is not possible to trap this error within dBASEIII+ - so you MUST ensure that printers are connected and switched on.

If the printer is merely off-line, the computer will simply hang until it is on-line.

2. All other errors should be trapped within dBASEIII+. A typical error will be when using the sampler version, and the number of records exceeds 31. A suitable error message will be displayed on the status bar, and the program will continue normally, though, of course, the last record will be lost. Be particularly careful when setting up exercises using the sampler, that the cumulative total of orders does not exceed 31. Files should be cleared whenever you are given the opportunity, eg. after the audit trail.

Any error messages other than those relating to the size of files in the sampler version should be reported to the author.

Appendix B - Student Exercise

USING A COMPUTER FOR YOUR BUSINESS

INTRODUCTION

For the next two lessons (3 hours) you will be using a computer as you would use it in a 'real' business. The software you will be using - the Order Processing Package - is similar to the software you could buy commercially, except that it was written especially for use in education. It is designed to be simple to understand and easy to use. Throughout the package, there are help screens available which will explain what you are doing.

The exercise is in two parts:

- first, you will work through a list of tasks on the computer. Instructions for this are given on the next page;
- second, try to answer the questions on the question sheet which will be given out later. The answers to these questions can all be found in the help screens which you can call up in the package.

THE COMPANY

The company is a wholesale supplier of health foods to a variety of customers, including pubs, shops and restaurants. Products include crisps, nuts and raisins, rice, etc. The company is fairly small, so it hasn't got many customers yet. However, it has got a new computer, which is used to process orders and do the accounts. It is your job to do this.

Unfortunately, whoever installed the software (ie. whoever set it up for the company) put the wrong name on it - they called the

company "Jane's Boutique Ltd.", when its name is actually "Derby Health Foods Ltd." Your first job will be to correct this.

THE ORDER PROCESSING PACKAGE

The customers send in orders to the company, and these orders have to be processed - in other words, they have to be entered into the computer, invoices printed, etc. You also can get a number of reports from the computer, including customer statements (showing payments from, and invoices to, customers); stock valuations; lists of products, customers, and suppliers; etc.

Some of the information is confidential, so you need a password to get into the package at first. Once in the package, you may not be able to use all the facilities - for example, only someone with the highest level of access can get into the part which lets you put on new users, change their passwords, etc.

The package also lets you carry out certain vital procedures, such as making a backup of the data you enter, in case it is lost or damaged. This is explained in the help screens.

The exercise assumes that it is Friday afternoon, and you have to enter the orders that have arrived that day, as well as produce various reports required by the management.

USING THE PACKAGE

1. Once you have entered the package (you will be shown how to do this) read the information displayed at the start. This will tell you about the help screens, and how to select choices from the menus. You will then be asked to enter your USER ID (or identification). The bits you type are in **bold**. Enter:

MAVIS [and press return]

Now enter the password, which is actually the place she was born:

ECCLES [and return]

This will bring you to the main menu.

2. First, change the name of the company to the correct name - Derby Health Foods Ltd. Using the left or right arrow keys (or cursor control keys) move along to INSTALLATION and press return. A sub menu appears. Move down to CHANGE COMPANY DETAILS (using the up or down arrow keys) and press return. The existing company details will appear. Everything is OK except the name - so just type this over the existing name, and press return until the cursor moves to the bottom of the screen. You will return to the sub menu, and the name will change. Select the EXIT option - move down to it and press return - and the highlight moves to the main menu.

3. You now need to add a new customer. Select the MASTER FILES option, and then select CUSTOMER FILE. Look at the new menu that appears. The PREVIOUS and NEXT options take you backwards and forwards through the file, displaying the customers on the screen. You can also FIND customers using their shortname, ADD new customers, or CHANGE and DELETE customers.

First though, press the F1 key to see the help pages. This should appear on the screen. Read through them, to see exactly what a master file is for.

Now use the NEXT and PREVIOUS options to look at some of the customers.

When you are satisfied with this, select ADD, and enter the following details:

Short name: **CITY**

Full name: **CITY WHOLEFOODS LTD**

Address: 22 **MANSFIELD RD**

HYSON GREEN

Town: **NOTTINGHAM**

Postcode: **NG5 6AB**

Phone: **0602 34567**

Current balance: (don't enter anything)

Credit limit: **100**

Sales to date: (don't enter anything)

When you press return after the last entry, the screen will clear to enter another record. Simply press return again, to go back to the menu. Select EXIT to return to the master file sub menu; and EXIT again to return to the main menu.

4. It's probably a good idea for some groups to get printouts of the products and customers. You do not need to hand these in, so share the printouts between 2 or 3 groups. This is how you do it: Select REPORTS from the main menu. A new (reports) menu will appear. Move down to CUSTOMERS: LIST ALL CUSTOMERS TO PRINTER. After selecting this option, you are asked to choose P or X to print the report. You MUST make sure that you are connected to a

printer, that it is switched to you, and that it is on-line. Select P to print. When printing is complete, select X to return to the reports menu, and print out a product list in the same way. Make sure you ask for a list by product code.

Return to the reports menu, and then to the main menu.

5. You now need to enter some of the orders which have come in during the day. This is probably the hardest part of the program - if you make a mistake, or don't understand, ask for help.

Select ORDERS from the main menu, and read the help screen. Then select ENTER ORDERS - a new menu will appear. Again, read the help screen. This is particularly important, since it explains what batch processing is. When you have finished reading this, select BATCH and enter in the following data:

Number of items: 3

total of quantities: 160

These are the batch totals. After entering these, the screen changes so that you can enter the orders themselves. Enter the following data:

Order num: 106

Short name: **CR** [this will then display customers whose short name begins with CR - in this case, Crissy Foods Ltd. This is the correct name, so enter Y and return]

Description: **JORDANS** [again, you will be asked if this is the correct product - enter Y and return]

Enter quantity: 100

The cursor should now move back to 'description' - in case there are any more items for Crissy. In this case there are not, so press return ONCE, so you can enter the next order. This is

number 107, for Derek's Delicatessen, so enter:

107

DEREK

and Y for the correct name.

Now enter the following descriptions and quantities:

SOYA (for soya sausage)

40

MIX (for mixed nuts and raisins)

20

Now you have finished entering the batch - tap return until the entry box disappears. If you have entered the batch incorrectly a message will be printed saying this, and you should then re-enter it. If no message has been printed, the batch has been entered correctly.

This batch is actually very small, but some other orders have already been entered in to make this a bit more realistic. Select the LIST option to see them - but don't bother to print them out. Note that the full name of the products isn't displayed - but the product reference is. Use your list of products to see what the product name is.

6. The next stage is to process the orders you have entered (and those others previously entered). Return to the Orders and Invoices menu, and select PROCESS ORDERS. This simply updates the master files, eg. reducing the quantity in stock of each product by the amount of the order, after checking that there are sufficient stocks to meet the order. In this case, there is one order with insufficient stock - you can view this on the screen.

7. You can now produce the invoices. Select the option from the

menu - if you wish you can view them on the screen before printing out. You can only print out the invoices ONCE. Each group should get a printout of them, then return to the main menu.

8. Select BACKUP from the main menu. Look at the help pages to see why backups are needed. Then select B to actually backup the data files.

9. Finally, select the REPORTS option from the main menu. You should obtain the following reports: (where you have the option, don't get a printout - just view them on the screen. However, it is important that you understand the function of each report before moving onto the next. You should make brief notes about each report.)

- a list of customers over the credit limit
- customer statements
- re-order list of products
- look at the bar charts of products
- sales ledger
- and audit trail

You should now have some understanding of how to use the package. Now try the following tasks:

- make up some batches of orders, using the lists of customers and products, and enter them into the computer. Process them and produce invoices.
- experiment with the different options available - eg. look at the supplier file. Try to find your way around the package.
- try entering some payments by customers following the issuing of invoices.

- look at some of the other reports.

- and, finally, make sure you have seen all the help screens - you will need the information on them to do the next piece of work.

Appendix C - Question Sheet

QUESTIONS ON USING A COMPUTER IN YOUR BUSINESS

Answer these questions using the notes made when you used the order processing package. Most of the information needed is on the help screens.

1. Describe ONE method of stopping unauthorised people using the computer package.

2. Explain what a MASTER FILE is.

3. Explain what BATCH PROCESSING is, and give some of its advantages and disadvantages.

4. What is the alternative to batch processing? Give one application where this might be useful.

5. Explain what BACKUPS are, and give TWO situations where you might need to use your backup data.

6. What is HOUSEKEEPING (when used in a computer context)?

7. What is an AUDIT TRAIL, and why is it needed?

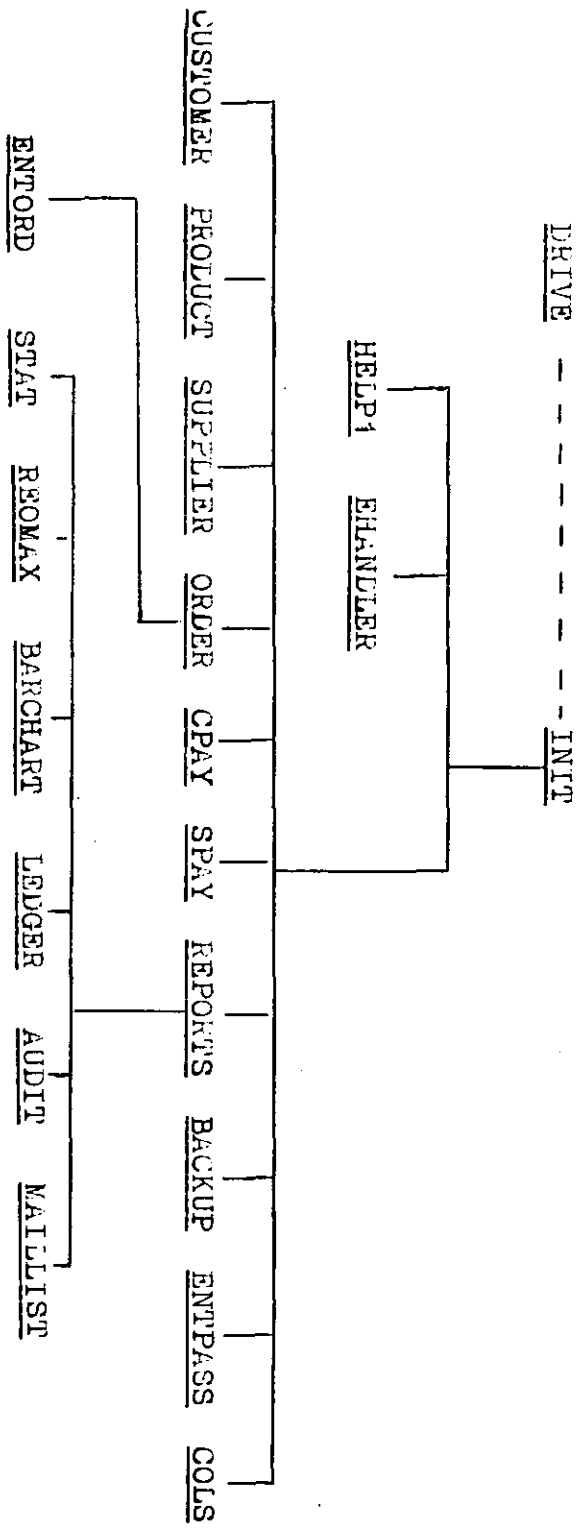
8. Why is a STOCK VALUATION report needed?

9. Why is it important to know which customers are spending over their credit limit?

10. Finally, write a few words about this particular package, eg.

- was it hard/easy to use?
- was there anything in it you didn't understand, and if so, what?
- did you find the instruction sheet hard/easy to follow?
- did you enjoy using the package or not?
- and add anything else you wish to say.

Appendix D - Module Tree



Appendix E - Filenames, Record Layouts

1. Database files, record layouts and index files.

All database files have the DBF extension. Index files have the NDX extension. The field types are C - character; N - numeric; L - logical; D - date .

a) CUSTOMER.DBF

Field	Field name	Type	Width
1	CUSTCODE	N	4
2	SHORTNAME	C	8
3	FULLNAME	C	20
4	ADD1	C	20
5	ADD2	C	20
6	ADD3	C	20
7	TOWN	C	15
8	PCODE	C	8
9	PHONE	C	15
10	BALANCE	N	9
11	CREDLIM	N	8
12	STDATE	N	10

Indexed on SHORTNAME to CNAME.NDX

CUSTCODE to CCODE.NDX

b) SUPPLIER.DBF

Field	Field name	Type	Width
1	SCUSTCODE	N	4
2	SSHORTNAME	C	8
3	SFULLNAME	C	20

4	SADD1	C	20
5	SADD2	C	20
6	SADD3	C	20
7	STOWN	C	15
8	SPCODE	C	8
9	SPHONE	C	15
10	SBALANCE	N	9
11	SCREDLIM	N	8
12	PTDATE	N	10

Indexed on SSHORTNAME to SNAME.NDX

SCUSTCODE to SCODE.NDX

c) PRODUCT.DBF

Field	**Field name**	**Type**	**Width**
1	REFERENCE	C	5
2	DESCRIP	C	20
3	GROUP	N	1
4	COSTPRICE	N	7
5	SALEPRICE	N	7
6	QIS	N	5
7	MINSTOCK	N	5
8	MAXSTOCK	N	5
9	ROQ	N	5
10	SUPPCODE	N	4

Indexed on DESCRIP to PDESC.NDX

REFERENCE to PREF.NDX

GROUP to PGROUP.NDX

SUPPCODE to PSUPP.NDX

d) ORDER.DBF

Field	Field name	Type	Width
1	CUSTCODE	N	4
2	REFERENCE	C	5
3	OQUANT	N	3
4	ODATE	D	8
5	ONUMBER	N	4
6	UPDATED	L	1
7	INVOICED	L	1
8	TRAILED	L	1
9	LEDGERED	L	1
10	INVNUM	N	5
11	INVDATE	D	8
12	ORDTOT	N	13

Indexed on ONUMBER to ONUM.NDX

CUSTCODE to OCUST.NDX

ODATE to ODATE.NDX

e) PAYMENT.DBF

Field	Field name	Type	Width
1	CUSTCODE	N	4
2	PDATE	D	8
3	AMOUNT	N	8
4	INVNUM	N	5
5	TRAILED	L	1

Indexed on CUSTCODE to PCODE.NDX

f) PRODHIST.DBF

Field	Field name	Type	Width
1	GROUP	N	1
2	SALES	N	5
3	SALEPRICE	N	7

Indexed on GROUP to HGROUP.NDX

g) PASS.DBF

Field	Field name	Type	Width
1	USER	C	8
2	PASSWORD	C	8
3	ACCESS	N	1

Indexed on USER TO UPASS.NDX

h) SCRATCH.DBF

Field	Field name	Type	Width
1	F1	C	20
1	F2	C	20
1	F3	C	20
1	F4	C	20
1	F5	C	20

i) There are also 4 database backup files, which have the same record layout as the original files, given in brackets. These are:

CUSTBACK.DBF (CUSTOMER)

PRODBACK.DBF (PRODUCT)

SUPPBACK.DBF (SUPPLIER)

ORDBACK.DBF (ORDER)

2. MEM files - ie. those used to store data such as the company name. All have the MEM extension.

a) COMPANY - stores company name, address etc.

b) INST and COLS - stores colours for screen displays.

c) DRIVE - stores data drive letter.

d) INV - stores current invoice number.

3. Label files - used for producing address labels - are CMAIL.LBL and SMAIL.LBL, for Customer and Supplier Files respectively.

4. INKEY.BIN is a binary file supplied by Ashton-Tate to overcome the bug in the INKEY() function, where the left arrow is not always trapped.

5. All other files have the PRG extension, and are program files. These are described in Section 7, and also appear in the module tree (Appendix D).

Appendix F - Listings

The listings appear on the subsequent pages, in the order in which they appear on the module tree, reading from left to right, and top to bottom.

```
*****************************************************************
*                                                               *
*                          DRIVE.PRG                            *
*                                                               *
*                              by                               *
*                                                               *
*                        Geoff Minshull                         *
*                                                               *
*                                                               *
*(c) copyright Geoff Minshull, 1987                             *
*****************************************************************

*
*  Called from: operating system, on entry to dBASEIII
*

*
*   This  program sets up the default data drive.  First time  of
* running package,  enter at system prompt,  dbase x:drive  -
* where x is drive with  data  disk.  Thereafter,  enter dbase
* x:init  to  start package.
*   Note  that  drive  reference is saved  on  dBASEIII+  disk  -
* therefore if a fresh disk is used, either run this routine
* again, or copy the
* file DRIVE.MEM to the new dBASEIII+ disk
*

CLEAR
@ 3,32 SAY 'SET DEFAULT DRIVE'
@ 4,32 SAY '------------------'
?  && blank lines
?
?
SET SAFETY OFF
SET TALK OFF
TEXT
    This  routine  allows you to specify the drive for  the  data
disk.  Normally
   on a twin floppy machine the dBASEIII+ disk would be in A, and
the data disk
   in B.  On a hard disk,  dBASEIII+  is likely to be on C   (the
hard disk)
   and the data disk in the floppy drive A.

    If  you wish,  you can copy all the files from the data  disk
floppy onto
   the  hard disk,  and use that for the dBASEIII+ programs,  AND
the files off
   the floppy.

    You can specify any drive for the data disk, from A to E.

 ENDTEXT

drlett=' '
```

```
DO WHILE drlett = ' '
 @ 19,3 SAY 'Enter the drive letter required for the data  files:
' GET drlett FUNCTION '!'
 READ
 IF AT(drlett,'ABCDEX')=0
  drlett=' '
  LOOP
 ENDIF
 STORE drlett+':'+'init.prg' TO sname
 IF .NOT. FILE(sname)
  @  21,3 SAY 'FILE DOES NOT EXIST!  Check that you have  entered
correct drive.'
  @ 22,3 SAY 'Re-enter drive letter - or X to quit'
  drlett=' '
  LOOP
 ENDIF

ENDDO

IF drlett='X'
  QUIT
ENDIF
SAVE  TO  DRIVE ALL LIKE DRLETT  && note that DRIVE is  saved  to
dBASEIII drive

CLEAR

TEXT

 The next time you use this program, you must enter at the MS-DOS
 system prompt - probably A> or C> -

  dbase x:init

  - where x is replaced by the data drive letter.

  Press any key to exit...

ENDTEXT

WAIT ''

QUIT
```

```
*****************************************************************
*                                                               *
*                          INIT.PRG                             *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called from: operating system when calling dBASE
*
*  This program  sets up a number of parameters at the start
*  Call it by a>dbase x:init where x is replaced
*  by the data drive reference. Note that first time of use,
*  enter a>dbase x:drive, to save drive reference.
*  The user then enters a password. If this is successfully
*  done, INIT then goes into the main menu routine.
*

*
*  First, check that data drive has been set up
*

IF .NOT. FILE('DRIVE.MEM')  && check that drive file exists
 CLEAR
 TEXT


 Drive file not present on dBASEIII+ disk.

 Start again from system prompt, and enter:

       dbase x:drive

[where x is data disk drive]


 ENDTEXT
 WAIT 'Press any key to exit dBASEIII+'
 QUIT

ENDIF

CLEAR ALL  && in case anything left open from different programs
CLEAR

SET FUNCTION 2 TO ''  && nul all function keys except F1
SET FUNCTION 3 TO ''
SET FUNCTION 4 TO ''
SET FUNCTION 5 TO ''
SET FUNCTION 6 TO ''
```

```
SET FUNCTION 7 TO ''
SET FUNCTION 8 TO ''
SET FUNCTION 9 TO ''
SET FUNCTION 10 TO ''

*
* set up system variables and parameters
*

RESTORE FROM DRIVE ADDITIVE
SET DEFAULT TO &drlett
PUBLIC
col1,col2,col3,col4,comname,comadd1,comtown,compcode,comphone,comva

&& make variables global
LOAD INKEY   && to avoid dbaseIII bug on leftarrow trap
SET TALK OFF
SET SAFETY OFF  && allow files to be overwritten without warning
SET STATUS OFF
SET SCOREBOARD OFF
@ 11,21 SAY 'Please wait whilst files are set up.....'
SET TYPEAHEAD TO 1 && reduce keyboard buffer
RESTORE FROM COLS ADDITIVE  && get colours
RESTORE FROM COMPANY ADDITIVE  && get company details
SET COLOR TO &col1/&col2
SET EXACT OFF  && switch off exact match for char strings
SET INTENSITY OFF  && switch off highlighted data entry
SET BELL OFF
SET CONFIRM ON     && need return when field filled
SET DATE BRITISH   && dd/mm/yy format
ON ERROR DO EHANDLER  && error handling routine

* open files

SET PROCEDURE TO HELP1
SELECT 1
USE CUSTOMER INDEX CNAME,CCODE ALIAS CUSTOMER
SELECT 2
USE PRODUCT ALIAS PRODUCT

*
*  This routine prints out the title screens, gets and validates
* the password.
*

CLEAR
@ 10,24 SAY 'ORDER PROCESSING USING A COMPUTER'
@ 12,39 SAY 'by'
@ 14,33 SAY 'Geoff Minshull'
@ 23,1 SAY 'Copyright: Geoff Minshull, 1987'

READ  && wait for keypress
CLEAR
@ 2,40-(LEN(comname)*.5) SAY comname
TEXT
```

```
  This suite of programs is written using dBASEIII+.  The package
provides
some of the basic operations of a computerised accounting system,
including:

      - updating files on customers, suppliers and products
      - processing orders (batch and individually)
      - printing invoices
      - printing address lists
      - extensive reporting facilities
      - colour customisation
      - etc.

  It  has been designed to be as easy to use as possible,  whilst
at the same
time  providing  a good introduction to a  computerised  accounts
system. At
various  points  within the program HELP screens  are  available,
which explain
what is happening at that particular point.  Pressing the F1  key
at the top
left of the keyboard will give you the HELP screen if available.

  When you are ready to continue, press any key.....
ENDTEXT

bar='introduction'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2
READ
CLEAR

TEXT

  Finding your way around the program is easy.

  Usually,  there will be a choice of options,  either across the
top of the
screen or down the left hand side. Selecting an option from these
menus is
done  by moving the highlight to the option you want,  using  the
cursor control
keys (the arrow keys) and pressing the RETURN key.

  You  can also use the HOME and END keys to move quickly to  the
first and
last options on the menus.

  DO NOT repeatedly tap the keys, or hold them down - it will not
make
the program work any faster,  but you may end up somewhere in the
program
```

```
where you don't want to be!

  At the very bottom of the screen is the STATUS BAR - this will
tell you
which part of the program you are in, if you are not sure. It
also says
if there is a HELP screen available; and it gives you the current
date.


  When you are ready to continue, press any key....
ENDTEXT

bar='introduction'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2
READ
CLEAR

TEXT

  Before you can start the program, you have to enter a password,
which
should have been given to you by your lecturer.

  You have 3 attempts to enter it. If you do not enter a valid
password
the program will abort.

  The purpose of a password is to prevent unauthorised people
using the
package.

  It also is a way of restricting which parts of the package you
can use.
Some people will be able to use all the functions - others will
only be
able to use some of them. So, for example, a computer operator
using
a payroll package might be able to enter in details of hours
worked by
the staff - but might not be allowed to see what rate per hour
they are
actually paid.


  Press any key to start entering your USER ID (identification)
and password.
ENDTEXT
```

```
bar='introduction'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2
READ

*
*  now get password
*

SELECT 10 && use as temporary work area
USE PASS INDEX UPASS
CLEAR
@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE('-',LEN(comname))
bar='get password'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

suser=SPACE(8)
ecount=0  && error count
looping=.T.

DO WHILE looping
 suser=SPACE(8)
 @ 11,21 SAY SPACE(15)
 @  9,5  SAY  'Enter USER ID:  ' GET suser FUNCTION  '!'  PICTURE
'AAAAAAAA'
 READ
 IF READKEY()=36
  DO HENTPASS
  LOOP
 ENDIF
 column=21
 @ 11,5 SAY 'Enter PASSWORD: '
 I=0
 mpass=' '

 DO WHILE I<>13
  SET CONSOLE OFF
  I=0
  DO WHILE I=0
   I=INKEY()
  ENDDO
  IF I=28
   DO HENTPASS
   LOOP
  ENDIF
```

```
  IF I=13
   LOOP
  ENDIF
  SET CONSOLE ON
  @ 11,column SAY '*'
  column=column+1
  IF column=31  && to restrict entry field
    I=13
    LOOP
  ENDIF
  mpass=mpass+CHR(I)
 ENDDO && I<>16
 SET CONSOLE ON
 mpass=LTRIM(mpass)
 mpass=UPPER(mpass)
 SEEK suser

 IF .NOT. FOUND() .OR. mpass<>RTRIM(PASSWORD)
  ecount=ecount+1
  IF ecount=3
   @ 15,5 SAY 'INVALID USER/PASSWORD! PROGRAM IS TERMINATING!'
   tcount=1
   DO WHILE tcount<300
    tcount=tcount+1
   ENDDO && tcount
   QUIT   && ie. back to operating system
  ENDIF  && ecount
  LOOP
 ENDIF  && not found
 looping=.F.
ENDDO  && looping

PUBLIC acclev
acclev=ACCESS  &&  access level for routines

* close file

USE
SELECT 3
USE ORDER INDEX ONUM,OCUST ALIAS ORDER

*
*
*                       NOW DO MAIN MENU ROUTINE
*
*

CLEAR


menuck=.T.

* selection number

sel=1
```

```
I=13

DO WHILE menuck

 IF I=13
  @ 1,40-(LEN(comname)*.5) SAY comname
  @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
  bar='main menu'
  SET COLOR TO &col2/&col1
  @ 24,0 SAY SPACE(80)
  @ 24,1 SAY '<F1> for help'
  @ 24,40-(LEN(bar)*.5) SAY bar
  @ 24,71 SAY DATE()
  SET COLOR TO &col1/&col2
 ENDIF  && I=13

 * column location of menu items in variable line

 STORE '01162536465570' TO locline

 * length of items in locline

 STORE '12060807061204' TO lenline

 STORE  'Master files    Orders    Payments    Reports    Backup
Installation   Exit' TO line
 STORE 'Use customer, product and supplier files         ' TO msg1
 STORE 'Process orders, print invoices                   ' TO msg2
 STORE 'Payments to suppliers and from customers         ' TO msg3
 STORE 'Produce various reports                          ' TO msg4
 STORE 'Backup and restore data files                    ' TO msg5
 STORE 'Change users, colours, company name, clear files' TO msg6
 STORE 'Exit from the program                            ' TO msg7

 SET COLOR TO &col1/&col2
 @ 4,1 SAY line
 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(locline,sel*2-1,2))
 hilite=SUBSTR(line,VAL(SUBSTR(locline,sel*2-1,2)),;
                    VAL(SUBSTR(lenline,sel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'msg' + STR(sel,1) TO msgnum
 @ 5,1 SAY &msgnum
 @ 6,0 CLEAR TO 23,79

  I=0

  X=CHR(200)
  DO WHILE X=CHR(200)  && routine necessary because left arrow
   CALL INKEY WITH X   && not trapped by INKEY()
  ENDDO
  I=ASC(X)
```

```
DO CASE

 CASE I=28
  DO HMAINMEN

 CASE I=1  && home key
  sel=1

 CASE I=6  && end key
  sel=7

 CASE I=4  && right arrow
  IF sel=7
    sel=1
   ELSE
    sel=sel+1
  ENDIF

 CASE I=19  && left arrow
  IF sel=1
    sel=7
   ELSE
    sel=sel-1
  ENDIF

 CASE I=13  && return

  DO CASE

   CASE sel=1  && master files
    mmenuck=.T.
    msel=1
    mlstsel=4  && number of choices

    STORE 'Customer file' TO mmsg1
    STORE 'Product file' TO mmsg2
    STORE 'Supplier file' TO mmsg3
    STORE 'Exit' TO mmsg4

    SET COLOR TO &col2/&col1
    @ 12,3 SAY mmsg1
    SET COLOR TO &col1/&col2
    @ 13,3 SAY mmsg2
    @ 14,3 SAY mmsg3
    @ 15,3 SAY mmsg4

    bar='master files menu'
    SET COLOR TO &col2/&col1
    @ 24,0 SAY SPACE(80)
    @ 24,1 SAY '<F1> for help'
    @ 24,40-(LEN(bar)*.5) SAY bar
    @ 24,71 SAY DATE()
    SET COLOR TO &col1/&col2

    DO WHILE mmenuck
```

```
        STORE 'mmsg' + STR(mlstsel,1) TO mmsgnum
        @ mlstsel+11,3 SAY &mmsgnum
        STORE 'mmsg' + STR(msel,1) TO mmsgnum
        SET COLOR TO &col2/&col1
        @ msel+11,3 SAY &mmsgnum
        SET COLOR TO &col1/&col2
*
* clear keyboard buffer
*
        SET TYPEAHEAD TO 0
        SET TYPEAHEAD TO 1

        I=0
        DO WHILE I=0
         I=INKEY()    && inkey ok here because leftarrow not used
        ENDDO

        mlstsel=msel

        DO CASE

         CASE I=28
          DO HMASTMEN

         CASE I=1
          msel=1

         CASE I=6
          msel=4

         CASE I=24  && down arrow
          IF msel=4
           msel=1
           ELSE
           msel=msel+1
          ENDIF

         CASE I=5  && up arrow
          IF msel=1
           msel=4
           ELSE
           msel=msel-1
          ENDIF

         CASE I=13

         DO CASE

          CASE msel=1  && customer
           DO CUSTOMER
           I=13  && to make sure screen is redrawn

          CASE msel=2  && product
           DO PRODUCT
```

```
          l=13

         CASE msel=3  && supplier
          DO SUPPLIER
          l=13

         CASE msel=4   && exit option
          mmenuck=.F.
         ENDCASE
*
*
        CASE l=27
         mmenuck=.F.

        OTHERWISE
         ? CHR(7)
        ENDCASE

        IF l=13 .AND. msel<>4
*  restore  screen -- use for hormen called  from  vermen  within
* hormen
* called prog needs to have renamed msg,line,locline,loclen
          sel=1  &&  ie. number of calling sel
          menuck=.T.
          @ 0,0 CLEAR TO 23,79
          @ 1,40-(LEN(comname)*.5) SAY comname
          @     2,40-(LEN(comname)*.5)    SAY    REPLICATE    ('-
',LEN(comname))
          SET COLOR TO &col1/&col2
          @ 4,1 SAY line
          SET COLOR TO &col2/&col1
          col=VAL(SUBSTR(locline,sel*2-1,2))
          hilite=SUBSTR(line,VAL(SUBSTR(locline,sel*2-1,2)),;
                    VAL(SUBSTR(lenline,sel*2-1,2)))
          @ 4,col SAY hilite
          SET COLOR TO &col1/&col2
          STORE 'msg' + STR(sel,1) TO msgnum
          @ 5,1 SAY &msgnum
          @ 12,3 SAY mmsg1
          @ 13,3 SAY mmsg2
          @ 14,3 SAY mmsg3
          @ 15,3 SAY mmsg4
          bar='master files menu'
          SET COLOR TO &col2/&col1
          @ 24,0 SAY SPACE(80)
          @ 24,1 SAY '<F1> for help'
          @ 24,40-(LEN(bar)*.5) SAY bar
          @ 24,71 SAY DATE()
          SET COLOR TO &col1/&col2
         ENDIF  && l=13

        ENDDO  && mmenuck
        @ 6,0 CLEAR TO 23,79

      CASE sel=2 && orders
```

```
 DO ORDER

CASE sel=3 && payments
 mmenuck=.T.
 msel=1
 mlstsel=3  && number of choices

 STORE 'Payments from customers' TO mmsg1
 STORE 'Payments to suppliers' TO mmsg2
 STORE 'Exit' TO mmsg3

 SET COLOR TO &col2/&col1
 @ 12,3 SAY mmsg1
 SET COLOR TO &col1/&col2
 @ 13,3 SAY mmsg2
 @ 14,3 SAY mmsg3
 bar='payments menu'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,1 SAY '<F1> for help'
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2

 DO WHILE mmenuck

  STORE 'mmsg' + STR(mlstsel,1) TO mmsgnum
  @ mlstsel+11,3 SAY &mmsgnum
  STORE 'mmsg' + STR(msel,1) TO mmsgnum
  SET COLOR TO &col2/&col1
  @ msel+11,3 SAY &mmsgnum
  SET COLOR TO &col1/&col2
  I=0
  SET TYPEAHEAD TO 0  && clear buffer
  SET TYPEAHEAD TO 1

  DO WHILE I=0
   I=INKEY()   && inkey ok here because leftarrow not used
  ENDDO

  mlstsel=msel

  DO CASE

   CASE I=28
    DO HPAYMEN

   CASE I=1
    msel=1

   CASE I=6
    msel=3

   CASE I=24  && down arrow
    IF msel=3
```

```
            msel=1
            ELSE
            msel=msel+1
           ENDIF

          CASE I=5  && up arrow
           IF msel=1
            msel=3
            ELSE
            msel=msel-1
           ENDIF

          CASE I=13

          DO CASE

           CASE msel=1
            @ 5,1 SAY 'Work with Customer file                '
            DO CPAY
            I=13

           CASE msel=2
            @ 5,1 SAY 'Work with Supplier file                '
            DO SPAY
            I=13

           CASE msel=3
            mmenuck=.F.
           ENDCASE
*
*
          CASE I=27
           mmenuck=.F.
          OTHERWISE
           ? CHR(7)
          ENDCASE
* restore screen
          IF I=13 .AND. msel<>3
            @ 6,0 CLEAR TO 23,79
            @ 5,1 SAY &msgnum
            @ 12,3 SAY mmsg1
            @ 13,3 SAY mmsg2
            @ 14,3 SAY mmsg3
            bar='payments menu'
            SET COLOR TO &col2/&col1
            @ 24,0 SAY SPACE(80)
            @ 24,1 SAY '<F1> for help'
            @ 24,40-(LEN(bar)*.5) SAY bar
            @ 24,71 SAY DATE()
            SET COLOR TO &col1/&col2
          ENDIF  && I=13
         ENDDO  && mmenuck
          @ 6,0 CLEAR TO 23,79

       CASE sel=4  && reports
```

```
 IF acclev<2
  @ 18,3 SAY 'ACCESS NOT AUTHORISED! Press any key...'
  READ
  @ 18,3 SAY SPACE(39)
 ELSE
  DO REPORTS
 ENDIF

CASE sel=5  && backups
 IF acclev<3
  @ 18,3 SAY 'ACCESS NOT AUTHORISED! Press any key...'
  READ
  @ 18,3 SAY SPACE(39)
 ELSE
  DO BACKUP
  CLEAR
 ENDIF

CASE sel=6  && installation
 mmenuck=.T.
 msel=1
 mlstsel=5  && number of choices

 STORE 'Add/delete users' TO mmsg1
 STORE 'Change colours' TO mmsg2
 STORE 'Change company details' TO mmsg3
 STORE 'Clear old files' TO mmsg4
 STORE 'Exit' TO mmsg5
 SET COLOR TO &col2/&col1
 @ 12,3 SAY mmsg1
 SET COLOR TO &col1/&col2
 @ 13,3 SAY mmsg2
 @ 14,3 SAY mmsg3
 @ 15,3 SAY mmsg4
 @ 16,3 SAY mmsg5
 bar='installation menu'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,1 SAY '<F1> for help'
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2

 DO WHILE mmenuck


  STORE 'mmsg' + STR(mlstsel,1) TO mmsgnum
  @ mlstsel+11,3 SAY &mmsgnum
  STORE 'mmsg' + STR(msel,1) TO mmsgnum
  SET COLOR TO &col2/&col1
  @ msel+11,3 SAY &mmsgnum
  SET COLOR TO &col1/&col2
  l=0
  SET TYPEAHEAD TO 0
  SET TYPEAHEAD TO 1
```

```
DO WHILE I=0
 I=INKEY()    && inkey ok here because leftarrow not used
ENDDO

mlstsel=msel

DO CASE

 CASE I=28
  DO HINST

 CASE I=1
  msel=1

 CASE I=6
  msel=5

 CASE I=24  && down arrow
  IF msel=5
   msel=1
   ELSE
   msel=msel+1
  ENDIF

 CASE I=5  && up arrow
  IF msel=1
   msel=5
   ELSE
   msel=msel-1
  ENDIF

 CASE I=13

 DO CASE

  CASE msel=1  && change users, passwords etc
   IF acclev<>4
    @ 18,3 SAY 'ACCESS NOT AUTHORISED! Press any key...'
    READ
    @ 18,3 SAY SPACE(39)
    I=0  && avoid unnecesary screen redraw
   ELSE
    DO ENTPASS
    I=13
   ENDIF

  CASE msel=2  && change colours
   DO COLS
   I=13

  CASE msel=3  && change company details
   @ 6,0 CLEAR TO 23,79
   bar='change company details'
   SET COLOR TO &col2/&col1
   @ 24,0 SAY SPACE(80)
```

```
          @ 24,40-(LEN(bar)*.5) SAY bar
          @ 24,71 SAY DATE()
          SET COLOR TO &col1/&col2
          @  6,3  SAY 'Current details  displayed.  Type  in  new
details if required.'
          @ 8,1 TO 20,49
          comname=comname+SPACE(30-LEN(comname))
          comadd1=comadd1+SPACE(20-LEN(comadd1))
          comtown=comtown+SPACE(20-LEN(comtown))
          compcode=compcode+SPACE(9-LEN(compcode))
          comphone=comphone+SPACE(15-LEN(comphone))
          comvat=comvat+SPACE(15-LEN(comvat))
          @  9,3  SAY  'Company  name:   '  GET  comname  PICTURE
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
          @   11,3   SAY   'Address:   '  GET   comadd1   PICTURE
'XXXXXXXXXXXXXXXXXXXX'
          @   13,3   SAY   'Town:    '   GET   comtown    PICTURE
'XXXXXXXXXXXXXXXXXXXX'
          @   15,3   SAY  'Postcode:   '  GET  compcode   PICTURE
'XXXXXXXXX'
          @   17,3  SAY  'Phone  no:   '  GET  comphone   PICTURE
'XXXXXXXXXXXXXXX'
          @   19,3   SAY   'VAT  no:   '   GET   comvat   PICTURE
'###############'
          READ
          comname=RTRIM(comname)
          comadd1=RTRIM(comadd1)
          comtown=RTRIM(comtown)
          compcode=RTRIM(compcode)
          comphone=RTRIM(comphone)
          comvat=RTRIM(comvat)
          SAVE TO COMPANY ALL LIKE com*
          invoice=0
          SAVE TO INV ALL LIKE invoice && reset inv.  no. for new
company
          l=13

         CASE msel=4   && zap files
          IF acclev<4
           @ 18,3 SAY 'ACCESS NOT AUTHORISED! Press any key...'
           READ
           @ 18,3 SAY SPACE(39)
           LOOP
          ENDIF
          choice=' '
          DO WHILE choice<>'X'
          choice=' '
           @ 6,0 CLEAR TO 23,79
           bar='clear files'
           SET COLOR TO &col2/&col1
           @ 24,0 SAY SPACE(80)
           @ 24,40-(LEN(bar)*.5) SAY bar
           @ 24,71 SAY DATE()
           SET COLOR TO &col1/&col2
           @  7,3  SAY 'All the master files  should  be  cleared
```

```
using the delete options'
          @  8,2 SAY 'provided.  However,  you may also want  to
clear certain other files.'
          @  9,2  SAY 'This option clears  the  backup,  product
sales history, and the'
          @ 10,2 SAY 'customer payment files.'
          @  12,2 SAY 'USE THIS OPTION WITH CARE.  IF IN  DOUBT,
BACKUP YOUR DISK FIRST!'
          @ 14,2 SAY 'Clear all Backup files      - B'
          @ 15,2 SAY 'Clear product History file  - H'
          @ 16,2 SAY 'Clear customer Payment file - P'
          @ 17,2 SAY '                            eXit - X'
         DO   WHILE   choice<>'X'   .AND.    choice<>'B'   .AND.
choice<>'H' .AND. choice<>'P'
          @ 14,35 SAY 'Enter your choice:  ' GET choice FUNCTION
'!'
          READ
         ENDDO
         SELECT 10

         DO CASE

          CASE choice='B'
           @ 19,2 SAY 'Deleting.....      '
           USE CUSTBACK
           ZAP
           USE PRODBACK
           ZAP
           USE SUPPBACK
           ZAP
           USE ORDBACK
           ZAP
           @ 19,2 SAY 'Backups deleted.         '

          CASE choice='H'
           @ 19,2 SAY 'Deleting.....      '
           USE PRODHIST INDEX HGROUP
           ZAP
           @ 19,2 SAY 'Product history deleted.'

          CASE choice='P'
           @ 19,2 SAY 'Deleting.....      '
           USE PAYMENT INDEX PCODE
           ZAP
           @ 19,2 SAY 'Payments deleted.        '
          CASE choice='X'
           LOOP
          ENDCASE
         ENDDO  && choice<>'x'

         @ 6,0 CLEAR TO 23,79
         SELECT 10
         USE

        CASE msel=5   && exit option
```

```
          mmenuck=.F.
         ENDCASE
*
*
        CASE I=27
         mmenuck=.F.
        OTHERWISE
         ? CHR(7)
        ENDCASE
        IF I=13 .AND. msel<>5
*  restore  screen  -- use for hormen called from  vermen  within
* hormen
* called prog needs to have renamed msg,line,locline,loclen
          sel=6  &&  ie. number of calling sel
          menuck=.T.
          @ 0,0 CLEAR TO 23,79
          @ 1,40-(LEN(comname)*.5) SAY comname
          @    2,40-(LEN(comname)*.5)    SAY    REPLICATE     ('-
',LEN(comname))
          SET COLOR TO &col1/&col2
          @ 4,1 SAY line
          SET COLOR TO &col2/&col1
          col=VAL(SUBSTR(locline,sel*2-1,2))
          hilite=SUBSTR(line,VAL(SUBSTR(locline,sel*2-1,2)),;
                  VAL(SUBSTR(lenline,sel*2-1,2)))
          @ 4,col SAY hilite
          SET COLOR TO &col1/&col2
          STORE 'msg' + STR(sel,1) TO msgnum
          @ 5,1 SAY &msgnum
          @ 12,3 SAY mmsg1
          @ 13,3 SAY mmsg2
          @ 14,3 SAY mmsg3
          @ 15,3 SAY mmsg4
          @ 16,3 SAY mmsg5
          bar='installation menu'
          SET COLOR TO &col2/&col1
          @ 24,0 SAY SPACE(80)
          @ 24,1 SAY '<F1> for help'
          @ 24,40-(LEN(bar)*.5) SAY bar
          @ 24,71 SAY DATE()
          SET COLOR TO &col1/&col2
         ENDIF  && I=13

       ENDDO  && mmenuck

    CASE sel=7  && exit package
      reply='N'
      @  7,1  SAY  'Exit  to operating system  Y/N  '  GET  reply
FUNCTION '!'
      READ
      IF reply='Y'
       QUIT
      ENDIF
      @ 7,1 SAY '                                    '
    ENDCASE
```

```
   OTHERWISE
    ? CHR(7)
  ENDCASE
ENDDO
```

```
*****************************************************************
*                                                               *
*                         HELP1.PRG                             *
*                                                               *
*                              by                               *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                               *
*****************************************************************

*
*  Called from: various routines, when F1 key pressed.
*

*
*  This program is a procedure file which contains the various
* help screens. These are called up with the F1 key - the name
* will be different at different parts of the package.
* Note only 32 procedures allowed in one file.
*


PROCEDURE HBACKUP

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'Making backups of data is essential. '
@ 10,41 SAY 'Disks can be damaged by heat, magnets'
@ 11,41 SAY 'dirt, fingerprints, etc. And if the  '
@ 12,41 SAY 'power fails, and the program is       '
@ 13,41 SAY 'interrupted, files can be lost. So    '
@ 14,41 SAY 'backups should be made regularly - if'
@ 15,41 SAY 'data changes frequently, you may need'
@ 16,41 SAY 'backups weekly, or even daily.        '
@ 17,41 SAY 'Normally, backups are made from one   '
@ 18,41 SAY 'floppy disk to another, or from a     '
@ 19,41 SAY 'hard disk to one or more floppies, or'
@ 20,41 SAY 'to a device called a tape streamer,   '
@ 21,41 SAY '        [press any key]               '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'which stores more data, but is a lot '
@ 10,41 SAY 'more expensive than floppies.         '
@ 11,41 SAY 'In this package, copies of the file   '
@ 12,41 SAY 'are made on the same disk. If the     '
@ 13,41 SAY 'original file is damaged, the data    '
@ 14,41 SAY 'can be recovered using the RESTORE    '
@ 15,41 SAY 'option, although the most recent      '
@ 16,41 SAY 'entries may be lost.                  '
@ 21,41 SAY '        [press any key]               '
READ
```

```
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN


PROCEDURE HMASTER

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'Maintaining and updating master files'
@ 10,41 SAY 'is an essential part of using a      '
@ 11,41 SAY 'computer at work. The master files   '
@ 12,41 SAY 'contain data which changes rarely,   '
@ 13,41 SAY 'such as addresses or product names;  '
@ 14,41 SAY 'or which is updated each time the    '
@ 15,41 SAY 'the program is run - eg. Quantity in '
@ 16,41 SAY 'Stock is stored in the Product Master'
@ 17,41 SAY 'file, but the actual figures will    '
@ 18,41 SAY 'change each time some are sold.      '
@ 19,41 SAY 'The usual operations on a master file'
@ 20,41 SAY 'include:                             '
@ 21,41 SAY '           [press any key]           '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9,41 SAY  ' - adding new records                '
@ 10,41 SAY ' - changing existing records         '
@ 11,41 SAY ' - deleting old records              '
@ 12,41 SAY ' - listing records                   '
@ 13,41 SAY ' - finding records which meet certain'
@ 14,41 SAY '   conditions.                       '
@ 21,41 SAY '           [press any key]           '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN

PROCEDURE HOAMEND
SET COLOR TO &col3/&col4
@ 7,45 CLEAR TO 18,79
@ 7,45 TO 18,79
@ 8, 46 SAY 'The last 3 options on this menu'
@  9,46 SAY 'allow you to change, delete and'
@ 10,46 SAY 'list orders. Changes and       '
@ 11,46 SAY 'deletions can only be made if  '
@ 12,46 SAY 'the master files have not yet  '
@ 13,46 SAY 'been updated. Orders can be    '
@ 14,46 SAY 'listed out at any time.        '
@ 15,46 SAY '                               '
@ 16,46 SAY '         [press any key]       '

READ
SET COLOR TO &col1/&col2
@ 7,45 CLEAR TO 18,79
```

```
RETURN


PROCEDURE HENTORD

SET COLOR TO &col3/&col4
@ 7,45 CLEAR TO 22,79
@ 7,45 TO 22,79
@ 8, 46 SAY 'Orders can be entered in 2 ways:'
@ 9, 46 SAY ' - one at a time, or in batches.'
@ 10,46 SAY 'Using the first method, orders  '
@ 11,46 SAY 'can be entered at any time - eg.'
@ 12,46 SAY 'as soon as they arrive in the   '
@ 13,46 SAY 'office. This means that the     '
@ 14,46 SAY 'master files are always right up'
@ 15,46 SAY 'to date - so, for example, the  '
@ 16,46 SAY 'Quantity in Stock will always be'
@ 17,46 SAY 'accurate because orders are     '
@ 18,46 SAY 'processed immediately.          '
@ 19,46 SAY 'However, in BATCH PROCESSING,   '
@ 20,46 SAY 'orders are processed in bundles,'
@ 21,46 SAY '       [press any key]          '
READ
@ 7,45 CLEAR TO 22,79
@ 7,45 TO 22,79
@ 8, 46 SAY 'probably at the end of the day. '
@ 9, 46 SAY 'Because of the delay between    '
@ 10,46 SAY 'receiving orders and processing '
@ 11,46 SAY 'them, the master files will be  '
@ 12,46 SAY 'out of date - so if they were   '
@ 13,46 SAY 'checked during the day to get   '
@ 14,46 SAY 'the current Quantity in Stock,  '
@ 15,46 SAY 'orders received would not have  '
@ 16,46 SAY 'been deducted - and so QIS would'
@ 17,46 SAY 'be wrong. Whether or not this is'
@ 18,46 SAY 'important depends on the actual '
@ 19,46 SAY 'application.  If ACCURACY is    '
@ 20,46 SAY 'more important than being right '
@ 21,46 SAY '       [press any key]          '
READ
@ 7,45 CLEAR TO 22,79
@ 7,45 TO 22,79
@ 8, 46 SAY 'up to date, then batching may be'
@ 9, 46 SAY 'better. In this package, when   '
@ 10,46 SAY 'entering batches, the operator  '
@ 11,46 SAY 'first has to enter the total    '
@ 12,46 SAY 'number of items  (batch total)  '
@ 13,46 SAY 'and the total of the quantities '
@ 14,46 SAY 'of all the orders in that batch '
@ 15,46 SAY '(hash total). These are worked  '
@ 16,46 SAY 'out beforehand, by the data     '
@ 17,46 SAY 'preparation staff. When the     '
@ 18,46 SAY 'orders are entered, the computer'
@ 19,46 SAY 'also carries out the same sums. '
@ 20,46 SAY 'If the computer totals do not   '
```

```
@ 21,46 SAY '       [press any key]          '
READ
@ 7,45 CLEAR TO 22,79
@ 7,45 TO 22,79
@ 8, 46 SAY 'match those entered by the      '
@ 9, 46 SAY 'operator, then something is     '
@ 10,46 SAY 'wrong - eg. a quantity entered  '
@ 11,46 SAY 'as 64 instead of 46. If there is'
@ 12,46 SAY 'an error, the batch has to be   '
@ 13,46 SAY 'checked and re-entered.         '
@ 14,46 SAY 'In this way, batching gives more'
@ 15,46 SAY 'accurate input.                 '
@ 16,46 SAY 'In this package, keep your      '
@ 17,46 SAY 'batches small - 10 at the most -'
@ 18,46 SAY 'in case you have to re-enter the'
@ 19,46 SAY 'orders!                         '
@ 21,46 SAY '       [press any key]          '
READ
SET COLOR TO &col1/&col2
@ 7,45 CLEAR TO 22,79
RETURN

PROCEDURE HREPORTS

SET COLOR TO &col3/&col4
@ 4,48 CLEAR TO 22,79
@ 4,48 TO 22,79
@ 6, 49 SAY 'A vital function of any       '
@ 7, 49 SAY 'information processing system '
@ 8, 49 SAY 'is the reports which it can   '
@ 9, 49 SAY 'produce. These reports are    '
@ 10, 49 SAY 'needed to give important      '
@ 11, 49 SAY 'information for the efficient '
@ 12,49 SAY 'running of the business.      '
@ 13,49 SAY 'Eg. the first report shown is '
@ 14,49 SAY 'a list of customers who owe   '
@ 15,49 SAY 'more than their credit limit. '
@ 16,49 SAY 'This is useful for maintaining'
@ 17,49 SAY 'a good cash flow, as well as  '
@ 18,49 SAY 'making sure customers do not  '
@ 19,49 SAY 'go too far into debt.         '
@ 21,49 SAY '      [press any key]         '
READ
SET COLOR TO &col1/&col2
@ 4,48 CLEAR TO 22,79
RETURN

PROCEDURE HAUDIT

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'An audit trail provides a record of  '
```

```
@ 10,41 SAY 'the transactions processed by the    '
@ 11,41 SAY 'system. In accounts kept manually, it'
@ 12,41 SAY 'is possible to go backwards through  '
@ 13,41 SAY 'the various books to trace the path  '
@ 14,41 SAY 'of a particular transaction. This    '
@ 15,41 SAY 'cannot be easily or cheaply done in a'
@ 16,41 SAY 'computerised system. But auditors    '
@ 17,41 SAY 'will still need to be able to trace  '
@ 18,41 SAY 'transactions. Hence the need for the '
@ 19,41 SAY 'AUDIT TRAIL.                         '
@ 20,41 SAY 'An audit trail is basically a print  '
@ 21,41 SAY '         [press any key]             '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'out of transactions which have been  '
@ 10,41 SAY 'processed. In this package, the audit'
@ 11,41 SAY 'trail is in 2 parts. The first part  '
@ 12,41 SAY 'is a list of orders and the related  '
@ 13,41 SAY 'invoice number; the second part lists'
@ 14,41 SAY 'payments made and received. After    '
@ 15,41 SAY 'this, you can delete old records from'
@ 16,41 SAY 'the various files - normally, this   '
@ 17,41 SAY 'should be done. Records only appear  '
@ 18,41 SAY 'once in the audit trail - and they   '
@ 19,41 SAY 'only appear AFTER an invoice has been'
@ 20,41 SAY 'printed.                             '
@ 21,41 SAY '         [press any key]             '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN


PROCEDURE HREOMAX

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'There are 2 reports concerned with   '
@ 10,41 SAY 'the amount of stock in the warehouse.'
@ 11,41 SAY 'One report deals with items where    '
@ 12,41 SAY 'stock is too low. The Product Master '
@ 13,41 SAY 'file stores the actual quantity in   '
@ 14,41 SAY 'stock; and the minimum amount below  '
@ 15,41 SAY 'which stock should not fall. The     '
@ 16,41 SAY 'report prints out those items where  '
@ 17,41 SAY 'stock levels have fallen below this  '
@ 18,41 SAY 'minimum amount. (Another option on   '
@ 19,41 SAY 'the reports menu prints out a letter '
@ 20,41 SAY 'to suppliers ordering more stock)    '
@ 21,41 SAY '         [press any key]             '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
```

```
@ 9, 41 SAY 'If stocks are too low, this can mean '
@ 10,41 SAY 'delays in meeting orders, and this   '
@ 11,41 SAY 'causes customer dissatisfaction and  '
@ 12,41 SAY 'ultimate loss of custom.             '
@ 13,41 SAY 'On the other hand, stocks which are  '
@ 14,41 SAY 'too high can also be a problem -     '
@ 15,41 SAY 'money is tied up unproductively in   '
@ 16,41 SAY 'stock. And, of course, the products  '
@ 17,41 SAY 'might not be selling any more - maybe'
@ 18,41 SAY 'they are clothes now out of fashion. '
@ 19,41 SAY 'A further problem with too much stock'
@ 20,41 SAY 'is perishable goods - there is little'
@ 21,41 SAY '       [press any key]               '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'point in stocking food past its      '
@ 10,41 SAY 'sell-by date. So another report is   '
@ 11,41 SAY 'produced - one which lists stock     '
@ 12,41 SAY 'above its maximum recommended level. '
@ 13,41 SAY '                                     '
@ 14,41 SAY 'Together, these 2 reports provide the'
@ 15,41 SAY 'basis of a system of stock control   '
@ 16,41 SAY 'which is an essential requirement    '
@ 17,41 SAY 'for running efficiently any business '
@ 18,41 SAY 'which holds stock, such as those     '
@ 19,41 SAY 'engaged in manufacturing and         '
@ 20,41 SAY 'distribution.                        '
@ 21,41 SAY '       [press any key]               '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN

PROCEDURE HENTPASS

SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'Security is very important in computer systems.'
@ 10,31 SAY 'Because information is so easily accessible, it'
@ 11,31 SAY 'is essential to make sure that only authorised '
@ 12,31 SAY 'people can see confidential files. A good     '
@ 13,31 SAY 'example is personnel files, which might hold  '
@ 14,31 SAY 'information about individuals which they would '
@ 15,31 SAY 'not want to be commonly known. Another example '
@ 16,31 SAY 'doctors records, which may well hold private,  '
@ 17,31 SAY 'and possibly embarassing, information. As well '
@ 18,31 SAY 'as making sure that actual physical access to  '
@ 19,31 SAY 'where the computer is located is restricted,   '
@ 20,31 SAY 'another common method of stopping unauthorised '
@ 21,31 SAY '            [press any key]                   '
READ
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
```

```
@ 9, 31 SAY 'use of a computer system is to install a        '
@ 10,31 SAY 'password routine. Only people with the right   '
@ 11,31 SAY 'password can use the computer. Cash dispensers '
@ 12,31 SAY 'outside banks use passwords; even some          '
@ 13,31 SAY 'telephones do now!.                             '
@ 14,31 SAY 'This package requires a password to enter it.  '
@ 15,31 SAY 'However, different passwords give access to     '
@ 16,31 SAY 'different parts of the package - so not         '
@ 17,31 SAY 'everyone can get a report, for example.         '
@ 18,31 SAY 'Passwords should be used carefully, and not     '
@ 19,31 SAY 'given to others.If your cash dispenser password'
@ 20,31 SAY 'is stolen, you may soon find you have no money!'
@ 21,31 SAY '              [press any key]                   '
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN


PROCEDURE HPAYMEN

SET COLOR TO &col3/&col4
@ 8,37 CLEAR TO 22,79
@ 8,37 TO 22,79
@ 9, 38 SAY 'This submenu is concerned with the       '
@ 10,38 SAY 'processing of payments received by the   '
@ 11,38 SAY 'company as a result of invoices sent out'
@ 12,38 SAY '(payments from customers option); and    '
@ 13,38 SAY 'payments made by the company for goods   '
@ 14,38 SAY 'bought (payments to suppliers). The      '
@ 15,38 SAY 'processing includes updating the balance'
@ 16,38 SAY 'field in the Customer and Supplier       '
@ 17,38 SAY 'master files.                            '
@ 18,38 SAY 'Payment details later appear on the      '
@ 19,38 SAY 'statements produced in the Reports menu.'
@ 20,38 SAY 'For a customer payment, you need to      '
@ 21,38 SAY '              [press any key]            '
READ
@ 8,37 CLEAR TO 22,79
@ 8,37 TO 22,79
@ 9, 38 SAY 'know the short name, the amount, and the'
@ 10,38 SAY 'invoice against which the payment is     '
@ 11,38 SAY 'made (though the last is not compulsory)'
@ 12,38 SAY 'For the payments to suppliers, you only '
@ 13,38 SAY 'need to know the short name and the      '
@ 14,38 SAY 'amount.                                  '
@ 15,38 SAY 'Keeping a close track of payments made   '
@ 16,38 SAY 'and received is a vital part of any      '
@ 17,38 SAY 'accounting system - payments received    '
@ 18,38 SAY 'are in fact the income of the company.   '
@ 19,38 SAY '                                         '
@ 20,38 SAY '                                         '
@ 21,38 SAY '              [press any key]            '
READ
SET COLOR TO &col1/&col2
```

```
@ 8,37 CLEAR TO 22,79
RETURN


PROCEDURE HMAINMEN

SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'This is the Main Menu. The various options on  '
@ 10,31 SAY 'it allow you to:                                '
@ 11,31 SAY '     -  update the 3 master files               '
@ 12,31 SAY '     -  work with orders                        '
@ 13,31 SAY '     -  process payments                        '
@ 14,31 SAY '     -  produce reports                         '
@ 15,31 SAY '     -  make backups of your data               '
@ 16,31 SAY '     -  change colours, passwords and the       '
@ 17,31 SAY 'company name; and clear old files.              '
@ 18,31 SAY 'Select an option by moving the highlight bar    '
@ 19,31 SAY 'with the cursor control (arrow) keys, and  then'
@ 20,31 SAY 'press the RETURN key for the option you want.   '
@ 21,31 SAY '              [press any key]                   '
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN

PROCEDURE HMASTMEN
SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'This is the Master File menu, which allows you '
@ 10,31 SAY 'to update the various Master Files. Each FILE -'
@ 11,31 SAY 'we will use the Customer File as an example -   '
@ 12,31 SAY 'consists of a SET of RELATED RECORDS. In other  '
@ 13,31 SAY 'words, the information stored about ONE         '
@ 14,31 SAY 'customer represents ONE RECORD in the file. All'
@ 15,31 SAY 'the records together make up the file. If the   '
@ 16,31 SAY 'customer details were all stored in one drawer  '
@ 17,31 SAY 'of a filing cabinet, that drawer would be the   '
@ 18,31 SAY 'same as the computer file. You would not - or   '
@ 19,31 SAY 'should not - store product records all mixed up'
@ 20,31 SAY 'with customer and supplier records in the same  '
@ 21,31 SAY '              [press any key]                   '
READ
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'drawer - the records in a particular file have  '
@ 10,31 SAY 'to be about the same sort of thing. Hence, they'
@ 11,31 SAY 'must be RELATED to each other in some way. In   '
@ 12,31 SAY 'this case, the records are all about customers.'
@ 13,31 SAY 'Each record  stores the same things for each    '
@ 14,31 SAY 'customer, although the actual data stored might'
@ 15,31 SAY 'be different. So there will be an address       '
@ 16,31 SAY 'stored for every customer, although the actual  '
```

```
@ 17,31 SAY 'address will not be the same. Other things      '
@ 18,31 SAY 'stored will include the credit limit, the       '
@ 19,31 SAY 'balance owed, and the telephone number. All     '
@ 20,31 SAY 'these things which are stored are called FIELDS'
@ 21,31 SAY '              [press any key]                   '
READ
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'In the Product Master file the fields will      '
@ 10,31 SAY 'include description; price; and quantity in     '
@ 11,31 SAY 'stock. Fields such as description will rarely,  '
@ 12,31 SAY 'if ever, change. Fields such as quantity in     '
@ 13,31 SAY 'stock will change every time some transactions  '
@ 14,31 SAY 'are processed. The field which stores the       '
@ 15,31 SAY 'balance owed by customers will also change      '
@ 16,31 SAY 'every time. So, in the same way as a file       '
@ 17,31 SAY 'consists of a set of related records, each      '
@ 18,31 SAY 'RECORD consists of a set of related fields.     '
@ 19,31 SAY 'Updating a master file simply means changing    '
@ 20,31 SAY 'the data in fields, and adding/deleting records.'
@ 21,31 SAY '              [press any key]                   '
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN

PROCEDURE HINST
SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'This sub-menu allows you to do a number of      '
@ 10,31 SAY 'different things.                               '
@ 11,31 SAY ' 1. Add/delete users. This is where you update  '
@ 12,31 SAY 'the file which contains the users and their     '
@ 13,31 SAY 'passwords. Access to this file is restricted,   '
@ 14,31 SAY 'so that unauthorised people cannot change       '
@ 15,31 SAY 'passwords.                                      '
@ 16,31 SAY ' 2. Change colours. This is where you can       '
@ 17,31 SAY 'change the colours used in the package.         '
@ 18,31 SAY ' 3. Change company details. This allows you to  '
@ 19,31 SAY 'change the name and address of the company used'
@ 20,31 SAY 'in this package.                                '
@ 21,31 SAY '              [press any key]                   '
READ
SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY ' 4. Clear old files. This option - to which     '
@ 10,31 SAY 'access is restricted - allows you to wipe old   '
@ 11,31 SAY 'files which are no longer required. This is     '
@ 12,31 SAY 'called HOUSEKEEPING, and is necessary so that   '
@ 13,31 SAY 'the disk where all the data is stored does not  '
@ 14,31 SAY 'get full up. You must be careful when deleting  '
@ 15,31 SAY 'old files. Once deleted, it is usually not      '
@ 16,31 SAY 'possible to recover them. So you must be sure   '
```

```
@ 17,31 SAY 'you really have finished with the files first. '
@ 18,31 SAY '                                               '
@ 19,31 SAY '                                               '
@ 20,31 SAY '                                               '
@ 21,31 SAY '              [press any key]                  '
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN

PROCEDURE HORDER
SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'This menu allows you to                        '
@ 10,31 SAY ' 1. Enter orders. This option is where you     '
@ 11,31 SAY 'enter orders received. It uses an ORDER FILE,  '
@ 12,31 SAY 'and you can also use this option to amend or   '
@ 13,31 SAY 'delete orders, as long as you have not yet     '
@ 14,31 SAY 'updated all the master files.                  '
@ 15,31 SAY ' 2. Use orders to update all files. This is    '
@ 16,31 SAY 'where the computer processes the orders against'
@ 17,31 SAY 'the master files - it checks to see if there is'
@ 18,31 SAY 'enough stock to meet an order, and prints out a'
@ 19,31 SAY 'list of items where stock is too low; and      '
@ 20,31 SAY 'updates the Quantity in Stock field. Orders    '
@ 21,31 SAY '              [press any key]                  '
READ
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'will NOT appear on an invoice until this option'
@ 10,31 SAY 'has been selected. Orders CANNOT be amended or '
@ 11,31 SAY 'deleted AFTER this option has been selected.   '
@ 12,31 SAY ' 3. Create and view/print invoices. This option'
@ 13,31 SAY 'actually produces the invoices. Make sure you  '
@ 14,31 SAY 'have processed the invoices first. When the    '
@ 15,31 SAY 'invoices are PRINTED, the balance field in the '
@ 16,31 SAY 'Customer Master file is updated.               '
@ 17,31 SAY '                                               '
@ 18,31 SAY '                                               '
@ 19,31 SAY '                                               '
@ 20,31 SAY '                                               '
@ 21,31 SAY '              [press any key]                  '
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN


PROCEDURE HMAIL
SET COLOR TO &col3/&col4
@ 8,37 CLEAR TO 22,79
@ 8,37 TO 22,79
@ 9, 38 SAY 'This menu is used to print out address  '
@ 10,38 SAY 'lists. Normally, this would be done onto'
```

```
@ 11,38 SAY 'special address label printer paper -    '
@ 12,38 SAY 'if this is available, you should put it '
@ 13,38 SAY 'in the printer now. Print out just one  '
@ 14,38 SAY 'label to test the paper position first. '
@ 15,38 SAY ' 1. Select which file you are using -   '
@ 16,38 SAY 'Customer or Supplier.                    '
@ 17,38 SAY ' 2. Choose how you are going to select  '
@ 18,38 SAY 'the addresses - ONE only of the options.'
@ 19,38 SAY ' 3. Select the print option. Addresses  '
@ 20,38 SAY 'can be viewed on the screen first.       '
@ 21,38 SAY '              [press any key]            '
READ
SET COLOR TO &col1/&col2
@ 8,37 CLEAR TO 22,79
RETURN

PROCEDURE HSTAT
SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'Statements should be produced          '
@ 10,41 SAY 'regularly and sent to the customer.    '
@ 11,41 SAY 'Sending statements out after the end   '
@ 12,41 SAY 'of the month is a good procedure. The  '
@ 13,41 SAY 'statement will tell the customer what  '
@ 14,41 SAY 'invoices have been sent to them, and   '
@ 15,41 SAY 'what payments have been received from  '
@ 16,41 SAY 'them. You should also keep a copy of   '
@ 17,41 SAY 'the statement in case the customer     '
@ 18,41 SAY 'contacts you with a query.             '
@ 19,41 SAY '                                       '
@ 20,41 SAY '                                       '
@ 21,41 SAY '         [press any key]               '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN

PROCEDURE HLEDGER
SET COLOR TO &col3/&col4
@ 8,30 CLEAR TO 22,79
@ 8,30 TO 22,79
@ 9, 31 SAY 'The Sales Ledger is a record of the sales made '
@ 10,31 SAY 'by the company (although this package does not '
@ 11,31 SAY 'have one, normally there would also be a       '
@ 12,31 SAY 'purchase ledger to record purchases). The      '
@ 13,31 SAY 'ledger gives monthly totals as well as a grand '
@ 14,31 SAY 'total of sales.                                '
@ 15,31 SAY 'A Sales Ledger is built up from the records    '
@ 16,31 SAY 'kept of sales made each day (in a manual       '
@ 17,31 SAY 'system this would be the Sales Day Book).      '
@ 18,31 SAY 'Once the files have been cleared (after the    '
@ 19,31 SAY 'audit trail has been produced) items will no   '
@ 20,31 SAY 'longer appear in the Sales Ledger.             '
@ 21,31 SAY '              [press any key]                  '
```

```
READ
SET COLOR TO &col1/&col2
@ 8,30 CLEAR TO 22,79
RETURN

PROCEDURE HPMAS

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'This files stores product records.   '
@ 10,41 SAY 'The REFERENCE NUMBER is a 5 character'
@ 11,41 SAY 'code - the first 3 must be letters,  '
@ 12,41 SAY 'the last 2 numbers. It is a good idea'
@ 13,41 SAY 'to make the codes meaningful, eg. a  '
@ 14,41 SAY '5lb. bag of beans could have the code'
@ 15,41 SAY 'BEA05. The PRODUCT GROUP is a number '
@ 16,41 SAY 'used to keep a track of similar types'
@ 17,41 SAY 'of product, eg. all home produced    '
@ 18,41 SAY 'goods could be in one group, all     '
@ 19,41 SAY 'imported goods in another.           '
@ 20,41 SAY 'The MINIMUM and MAXIMUM stock levels,'
@ 21,41 SAY '           [press any key]           '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9,41 SAY  'and the RE-ORDER QUANTITY are set by '
@ 10,41 SAY 'whoever is in charge of re-ordering. '
@ 11,41 SAY '                                     '
@ 12,41 SAY 'All the fields (ie. the entries)     '
@ 13,41 SAY 'should have some data entered.       '
@ 14,41 SAY '                                     '
@ 21,41 SAY '           [press any key]           '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN

PROCEDURE HSMAS

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'This files stores supplier records.  '
@ 10,41 SAY 'The SHORT NAME is used to find a     '
@ 11,41 SAY 'particular supplier, and means that  '
@ 12,41 SAY 'the only time you need to put in the '
@ 13,41 SAY 'full name is when entering a new     '
@ 14,41 SAY 'supplier.                            '
@ 15,41 SAY 'The CURRENT BALANCE shows how much   '
@ 16,41 SAY 'you owe the supplier. It is automat- '
@ 17,41 SAY 'ically updated when you order goods  '
@ 18,41 SAY 'or make payments.                    '
@ 19,41 SAY 'The CREDIT LIMIT is the amount of    '
@ 20,41 SAY 'credit the supplier will allow you.  '
```

```
@ 21,41 SAY '           [press any key]           '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9,41 SAY  'The PURCHASES TO DATE show how much  '
@ 10,41 SAY 'you have spent with this supplier.   '
@ 11,41 SAY '                                     '
@ 12,41 SAY '                                     '
@ 13,41 SAY '                                     '
@ 14,41 SAY '                                     '
@ 21,41 SAY '           [press any key]           '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN

PROCEDURE HCMAS

SET COLOR TO &col3/&col4
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9, 41 SAY 'This files stores customer records.  '
@ 10,41 SAY 'The SHORT NAME is used to find a     '
@ 11,41 SAY 'particular customer, and means that  '
@ 12,41 SAY 'the only time you need to put in the '
@ 13,41 SAY 'full name is when entering a new     '
@ 14,41 SAY 'customer.                            '
@ 15,41 SAY 'The CURRENT BALANCE shows how much   '
@ 16,41 SAY 'the customer owes you. It is automat-'
@ 17,41 SAY 'ically updated when you supply goods '
@ 18,41 SAY 'or the customer makes a payment.     '
@ 19,41 SAY 'The CREDIT LIMIT is the amount of    '
@ 20,41 SAY 'credit you will allow the customer.  '
@ 21,41 SAY '           [press any key]           '
READ
@ 8,40 CLEAR TO 22,79
@ 8,40 TO 22,79
@ 9,41 SAY  'The SALES TO DATE show how much the  '
@ 10,41 SAY 'customer has bought from you.        '
@ 11,41 SAY '                                     '
@ 12,41 SAY '                                     '
@ 13,41 SAY '                                     '
@ 14,41 SAY '                                     '
@ 21,41 SAY '           [press any key]           '
READ
SET COLOR TO &col1/&col2
@ 8,40 CLEAR TO 22,79
RETURN
```

```
*****************************************************************
*                                                               *
*                        EHANDLER.PRG                           *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called whenever an error occurs
*

*
*  This program handles all errors which occur, EXCEPT those
* connected with the operating system, eg. printer not
* connected, read/write error, etc. When an error occurs, an
* error message is printed on the status bar, after which
* processing can resume.
* Note that although this program HAS been tested, it is still
* likely that bugs will be found when it is used in the classroo
* by non-specialist teachers. This routine minimises any incon-
* venience which may be caused, and should mean that the lesson
* can proceed normally.
*

DO CASE

* this first error will only occur in sampler version of
* dBASE where file size is restricted.
*
 CASE ERROR()=5
  ebar='file full - no more records allowed'
  @ 24,40-(LEN(ebar)*.5) SAY ebar

 OTHERWISE

  ebar=MESSAGE()+'    '+STR(ERROR())    &&    ie.    dBASE    erro
message/number
  @ 24,40-(LEN(ebar)*.5) SAY ebar

 ENDCASE

READ
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

RETURN
```

```
*************************************************************
*                                                           *
*                      CUSTOMER.PRG                         *
*                                                           *
*                            by                             *
*                                                           *
*                      Geoff Minshull                       *
*                                                           *
*                                                           *
*©copyright Geoff Minshull, 1987                            *
*************************************************************

*
*  Called from: INIT
*
*  This program updates the master file.
*

CLEAR
SELECT CUSTOMER
GO TOP

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='customer file'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

menuck=.T.  && main loop
del=.F.  && pack only if deletions made

* selection number

sel=1
I=13  && to make sure record printed on first pass

* column location of menu items in variable line

STORE '01102332405161' TO loccline

* length of items in loccline

STORE '04080403060604' TO lencline

STORE  'Next     Previous     Find     Add     Change      Delete
Exit ' TO cline
STORE 'Find next customer                          ' TO cmsg1
STORE 'Find previous customer                      ' TO cmsg2
STORE 'Find customer by short name                 ' TO cmsg3
STORE 'Add new customer                            ' TO cmsg4
STORE 'Change customer details                     ' TO cmsg5
```

```
STORE 'Delete a customer                                    ' TO cmsg6
STORE 'Return to previous menu                              ' TO cmsg7

DO WHILE menuck

 @ 4,1 SAY cline

 IF I=13 .AND. (.NOT. EOF())
  @ 8,  1  SAY 'Short name: '
  @ 8, 13  SAY SHORTNAME
  @ 9,  1  SAY "Full name:"
  @ 9, 13  SAY FULLNAME
  @ 11,  1  SAY "Address: "
  @ 11, 11  SAY ADD1
  @ 12, 11  SAY ADD2
  @ 13, 11  SAY ADD3
  @ 14,  1  SAY "Town: "
  @ 14, 11  SAY TOWN
  @ 15,  1  SAY "Postcode: "
  @ 15, 11  SAY  PCODE
  @ 17,  1  SAY "Phone: "
  @ 17, 11  SAY PHONE
  @ 19,  1  SAY "Current Balance: "
  @ 19, 18 SAY SPACE(11)  && to blank out brackets
  @ 19, 18  SAY  BALANCE FUNCTION '(BZ'  && negative in brackets
  @ 20,1    SAY 'Credit Limit: '
  @ 20,18   SAY CREDLIM FUNCTION 'BZ'
  @ 21,1    SAY 'Sales to date: '
  @ 21,18   SAY STDATE FUNCTION 'BZ'
  @  7,  0  TO 22, 37

  IF DELETED()
   @ 6,1 SAY 'Record marked for deletion'
  ENDIF

 ENDIF


 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(loccline,sel*2-1,2))
 hilite=SUBSTR(cline,VAL(SUBSTR(loccline,sel*2-1,2)),;
                     VAL(SUBSTR(lencline,sel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'cmsg'+STR(sel,1) TO cmsgnum
 @ 5,1 SAY &cmsgnum

  I=0

* routine to deal with dbaseIII bug - not trapping leftarrow

  X=CHR(200)
  DO WHILE X=CHR(200)
   CALL INKEY WITH X
  ENDDO
```

```
I=ASC(X)
@ 6,1 SAY SPACE(27)
@ 23,1 SAY SPACE(41)

DO CASE

 CASE I=28  && F1 when inkey used
  DO HMASTER

 CASE I=1   && home
  sel=1

 CASE I=6   && end
  sel=7

 CASE I=4  && right arrow
  IF sel=7
    sel=1
   ELSE
    sel=sel+1
  ENDIF

 CASE I=19  && left arrow
  IF sel=1
    sel=7
   ELSE
    sel=sel-1
  ENDIF

 CASE I=13  && return
  DO CASE

   CASE sel=1  && next
    IF .NOT. EOF()
     SKIP
    ENDIF
    IF EOF()
     @ 23,1 SAY 'End of file - no more records!'
     GO BOTTOM
    ENDIF

  CASE sel=2  && previous
    IF .NOT. BOF()
     SKIP-1
    ELSE
     @ 23,1 SAY 'Start of file - no more records!'
    ENDIF

  CASE sel=3  && find
    dofind=.T.
    DO WHILE dofind
     ename=SPACE(8)
     @ 8,13 GET ename FUNCTION '!'
     @ 9,13 SAY SPACE (20)
     @ 11,11 SAY SPACE(20)
```

```
        @ 12,11 SAY SPACE(20)
        @ 13,11 SAY SPACE(20)
        @ 14,11 SAY SPACE(15)
        @ 15,11 SAY SPACE (8)
        @ 17,11 SAY SPACE(15)
        @ 19,18 SAY SPACE(8)
        @ 20,18 SAY SPACE(8)
        @ 21,18 SAY SPACE(15)
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HCMAS
         LOOP
        ENDIF
        IF ename=SPACE(8)
         dofind=.F.
         LOOP
        ENDIF
        ename=RTRIM(ename)
        SEEK ename
        IF .NOT. FOUND()
          @  23,1 SAY 'Customer does not exist.  Moved to start of
file'
          GO TOP
         ELSE
          @  23,1 SAY 'Customer found - select  NEXT  for  others.
'
        ENDIF
        dofind=.F.
       ENDDO   && dofind

     CASE sel=4 && add
       SET ORDER TO  && remove index to obtain last custcode
       adding =.T.
       pointer=RECNO()  && store current position
       DO WHILE adding
        ename=SPACE(8)
        @ 8,13 GET ename FUNCTION '!'
        @ 9,13 SAY SPACE(20)
        @ 11,11 SAY SPACE(20)
        @ 12,11 SAY SPACE(20)
        @ 13,11 SAY SPACE(20)
        @ 14,11 SAY SPACE(15)
        @ 15,11 SAY SPACE(8)
        @ 17,11 SAY SPACE(15)
        @ 19,18 SAY SPACE(8)
        @ 20,18 SAY SPACE(8)
        @ 21,18 SAY SPACE(15)
        READ

        IF READKEY()=36 .OR. READKEY()=292
         DO HCMAS
         LOOP
        ENDIF

        IF ename=SPACE(8)
```

```
          adding=.F.
          LOOP
         ENDIF
         GO BOTTOM
         code=CUSTCODE
         APPEND BLANK
         pointer=RECNO()  && to make sure record printed
         REPLACE CUSTCODE WITH code+1  && get unique key
         REPLACE SHORTNAME WITH ename
         nohelp=.t.
         DO WHILE nohelp
          nohelp=.f.
          @ 9,13 GET FULLNAME FUNCTION '!'
          @ 11,11 GET ADD1  FUNCTION '!'
          @ 12,11 GET ADD2 FUNCTION '!'
          @ 13,11 GET ADD3 FUNCTION '!'
          @ 14,11 GET TOWN FUNCTION '!'
          @ 15,11 GET PCODE FUNCTION '!'
          @ 17,11 GET PHONE FUNCTION '!'
* BXZ = left justify, zero as blank
          @ 19,18 GET BALANCE FUNCTION '(BZ' PICTURE '99999.99'
          @ 20,18 GET CREDLIM FUNCTION 'BZ' PICTURE '99999.99'
          @ 21,18 GET STDATE FUNCTION 'BZ' PICTURE '9999999.99'
          READ
          IF READKEY()=36 .OR. READKEY()=292
           DO HCMAS
           nohelp=.t.
           LOOP
          ENDIF
         ENDDO  && nohelp
        ENDDO  && adding
        SET ORDER TO 1   && restore index
        REINDEX
        GOTO pointer  && to make sure appended record onscreen

      CASE sel=5  && change
        nohelp=.t.
        DO WHILE nohelp
         nohelp=.f.
         @ 8,13 GET SHORTNAME FUNCTION '!'
         @ 9,13 GET FULLNAME FUNCTION '!'
         @ 11,11 GET ADD1 FUNCTION '!'
         @ 12,11 GET ADD2 FUNCTION '!'
         @ 13,11 GET ADD3 FUNCTION '!'
         @ 14,11 GET TOWN FUNCTION '!'
         @ 15,11 GET PCODE FUNCTION '!'
         @ 17,11 GET PHONE FUNCTION '!'
         @ 19,18 GET BALANCE FUNCTION '(BZ' PICTURE '99999.99'
         @ 20,18 GET CREDLIM FUNCTION 'BZ' PICTURE '99999.99'
         @ 21,18 GET STDATE FUNCTION 'BZ' PICTURE '9999999.99'
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HCMAS
          nohelp=.t.
          LOOP
```

```
        ENDIF
       ENDDO  && nohelp

     CASE sel=6  && delete
       DELETE
       del=.T.
       @ 23,1 SAY 'Customer deleted. Next one displayed'
       SKIP
       IF EOF()
        @ 23,1 SAY 'Deleted. Gone to start of file      '
        GO TOP
       ENDIF

     CASE sel=7  && exit
       menuck=.F.
     ENDCASE
    OTHERWISE
     ? CHR(7)
   ENDCASE

ENDDO && menuck

*
* pack at end to save processing time
*

IF del
  PACK
ENDIF
RETURN
```

```
*****************************************************************
*                                                               *
*                        PRODUCT.PRG                            *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*(c) copyright Geoff Minshull, 1987                             *
*****************************************************************

*
*  Called from: INIT
*

*
*  This program updates the master file.
*

CLEAR

SELECT PRODUCT
SET INDEX TO PDESC,PREF,PGROUP
GO TOP

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='product file'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

menuck=.T.
del=.F.  && pack only if deletions made

* selection number

sel=1
l=13  && to make sure record printed on first pass


* column location of menu items in variable line

STORE '01102332405161' TO locpline

* length of items in locpline

STORE '04080403060604' TO lenpline

STORE  'Next     Previous     Find     Add     Change     Delete
Exit ' TO pline
```

```
STORE 'Find next product                                    ' TO pmsg1
STORE 'Find previous product                                ' TO pmsg2
STORE 'Find product by full or partial reference            ' TO pmsg3
STORE 'Add new product                                      ' TO pmsg4
STORE 'Change product details                               ' TO pmsg5
STORE 'Delete a product                                     ' TO pmsg6
STORE 'Return to previous menu                              ' TO pmsg7

DO WHILE menuck

 @ 4,1 SAY pline

 IF l=13 .AND. (.NOT. EOF())
  @ 9,  1  SAY "Ref. number:"
  @ 9, 14  SAY REFERENCE
  @ 10,  1  SAY "Description:"
  @ 10, 14  SAY DESCRIP
  @ 11,  1  SAY 'Supp. code:'
  @ 11, 14  SAY SUPPCODE
  @ 12,  1  SAY "Product group:"
  @ 12, 17  SAY GROUP FUNCTION 'BZ'
  @ 14,  1  SAY "Cost Price:"
  @ 14, 14  SAY COSTPRICE  FUNCTION 'BZ'
  @ 15,  1  SAY "Sale price:"
  @ 15, 14  SAY SALEPRICE  FUNCTION 'BZ'
  @ 17,  1  SAY "Quantity in stock:"
  @ 17, 21  SAY QIS  FUNCTION 'BZ'
  @ 19,  1  SAY "Minimum Stock level:"
  @ 19, 23  SAY  MINSTOCK FUNCTION 'BZ'
  @ 20,  1  SAY "Maximum Stock level:"
  @ 20, 23  SAY  MAXSTOCK FUNCTION 'BZ'
  @ 21,  1  SAY "Reorder quantity:"
  @ 21, 23  SAY  ROQ FUNCTION 'BZ'
  @  8,  0  TO 22, 37

  IF DELETED()
   @ 7,1 SAY 'Record marked for deletion'
  ENDIF

 ENDIF


 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(locpline,sel*2-1,2))
 hilite=SUBSTR(pline,VAL(SUBSTR(locpline,sel*2-1,2)),;
                     VAL(SUBSTR(lenpline,sel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'pmsg'+STR(sel,1) TO pmsgnum
 @ 5,1 SAY &pmsgnum

  l=0

* routine to deal with dbaseIII bug - not trapping leftarrow
```

```
X=CHR(200)
DO WHILE X=CHR(200)
 CALL INKEY WITH X
ENDDO
I=ASC(X)

@ 7,1 SAY SPACE(27)
@ 23,1 SAY SPACE(40)

DO CASE

 CASE I=28
  DO HMASTER

 CASE I=1
  sel=1

 CASE I=6
  sel=7

 CASE I=4  && right arrow
  IF sel=7
    sel=1
   ELSE
    sel=sel+1
  ENDIF

 CASE I=19  && left arrow
  IF sel=1
    sel=7
   ELSE
    sel=sel-1
  ENDIF

 CASE I=13  && return

  DO CASE

   CASE sel=1  && next
    IF .NOT. EOF()
     SKIP
    ENDIF
    IF EOF()
     @ 23,1 SAY 'End of file - no more records!'
     GO BOTTOM
    ENDIF

   CASE sel=2  && previous
    IF .NOT. BOF()
     SKIP-1
    ELSE
     @ 23,1 SAY 'Start of file - no more records!'
    ENDIF

   CASE sel=3  && find
```

```
        dofind=.T.
        DO WHILE dofind
         eref=SPACE(5)
         @ 9,14 GET eref FUNCTION '!' PICTURE 'AAA99'
         @ 10,14 SAY SPACE (20)
         @ 11,14 SAY '     '
         @ 12,17 SAY '   '
         @ 14,14 SAY '         '
         @ 15,14 SAY '         '
         @ 17,21 SAY '       '
         @ 19,23 SAY '       '
         @ 20,23 SAY '       '
         @ 21,23 SAY '       '
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HPMAS
          LOOP
         ENDIF
         IF eref=SPACE(5)
          dofind=.F.
          LOOP
         ENDIF
         eref=RTRIM(eref)
         SEEK eref
         IF .NOT. FOUND()
           @  23,1 SAY 'Product does not exist.  Moved to start  of
file.'
           GO TOP
          ELSE
           @  23,1  SAY 'Product found - select  NEXT  for  other s.
'
         ENDIF
         dofind=.F.
        ENDDO   && dofind

      CASE sel=4  && add
        adding =.T.
        DO WHILE adding
         eref=SPACE(6)
         @ 9,14 GET eref FUNCTION '!' PICTURE 'AAA99'
         @ 10,14 SAY SPACE(20)
         @ 11,14 SAY '     '
         @ 12,17 SAY '   '
         @ 14,14 SAY '         '
         @ 15,14 SAY '         '
         @ 17,21 SAY '       '
         @ 19,23 SAY '       '
         @ 20,23 SAY '       '
         @ 21,23 SAY '       '
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HPMAS
          LOOP
         ENDIF
         IF eref=SPACE(6)
```

```
      adding=.F.
      LOOP
     ENDIF
     SEEK eref
     IF FOUND()
      @ 23,1 SAY 'Reference exists - press key to re-enter'
      WAIT ''
      LOOP
     ENDIF
     APPEND BLANK
     REPLACE REFERENCE WITH eref
     nohelp=.t.
     DO WHILE nohelp
      nohelp=.f.
      @ 10,14 GET DESCRIP FUNCTION '!'
      @ 11,14 GET SUPPCODE FUNCTION 'BZ' PICTURE '9999'
      @ 12,17 GET GROUP FUNCTION 'BZ' PICTURE '9'
      @ 14,14 GET COSTPRICE FUNCTION 'BZ'
      @ 15,14 GET SALEPRICE FUNCTION 'BZ'
      @ 17,21 GET QIS FUNCTION 'BZ'
      @ 19,23 GET MINSTOCK FUNCTION 'BZ'
      @ 20,23 GET MAXSTOCK FUNCTION 'BZ'
      @ 21,23 GET ROQ FUNCTION 'BZ'
      READ
      IF READKEY()=36 .OR. READKEY()=292
       DO HPMAS
       nohelp=.t.
       LOOP
      ENDIF
     ENDDO  && nohelp
    ENDDO  && adding

CASE sel=5  && change
    nohelp=.t.
    DO WHILE nohelp
     nohelp=.f.
     @ 10,14 GET DESCRIP FUNCTION '!'
     @ 11,14 GET SUPPCODE FUNCTION 'BZ' PICTURE '9999'
     @ 12,17 GET GROUP FUNCTION 'BZ' PICTURE '9'
     @ 14,14 GET COSTPRICE FUNCTION 'BZ'
     @ 15,14 GET SALEPRICE FUNCTION 'BZ'
     @ 17,21 GET QIS FUNCTION 'BZ'
     @ 19,23 GET MINSTOCK FUNCTION 'BZ'
     @ 20,23 GET MAXSTOCK FUNCTION 'BZ'
     @ 21,23 GET ROQ FUNCTION 'BZ'
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPMAS
      nohelp=.t.
      LOOP
     ENDIF
    ENDDO && nohelp

CASE sel=6  && delete
    DELETE
```

```
      del=.T.
      @ 23,1 SAY 'Product deleted. Next one displayed'
      SKIP
      IF EOF()
       @ 23,1 SAY 'Deleted. Gone to start of file      '
       GO TOP
      ENDIF

    CASE sel=7  && exit
      menuck=.F.
    ENDCASE
   OTHERWISE
    ? CHR(7)
  ENDCASE

ENDDO && menuck

IF del
  PACK
ENDIF
SET INDEX TO  && close indices

RETURN
```

```
*****************************************************************
*                                                               *
*                       SUPPLIER.PRG                            *
*                                                               *
*                            by                                 *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called from: INIT
*

*
*  This program updates the master file.
*

CLEAR

SELECT ORDER
USE  && close because too many files open
SELECT 10
USE SUPPLIER INDEX SNAME,SCODE ALIAS SUPPLIER

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='supplier file'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

menuck=.T.
del=.F.  && pack only if deletions made

* selection number

sel=1
l=13  && to make sure record printed on first pass

* column location of menu items in variable line

STORE '01102332405161' TO locsline

* length of items in locsline

STORE '04080403060604' TO lensline

STORE  'Next     Previous     Find     Add     Change     Delete
Exit ' TO sline
```

```
STORE 'Find next supplier                                    ' TO smsg1
STORE 'Find previous supplier                                ' TO smsg2
STORE 'Find supplier by short name                           ' TO smsg3
STORE 'Add new supplier                                      ' TO smsg4
STORE 'Change supplier details                               ' TO smsg5
STORE 'Delete a supplier                                     ' TO smsg6
STORE 'Return to previous menu                               ' TO smsg7

DO WHILE menuck

 @ 4,1 SAY sline

 IF I=13 .AND. (.NOT. EOF())
  @ 8,  1  SAY 'Short name: '
  @ 8, 13  SAY SSHORTNAME
  @ 9,  1  SAY "Full name:"
  @ 9, 13  SAY SFULLNAME
  @ 11,  1  SAY "Address: "
  @ 11, 11  SAY SADD1
  @ 12, 11  SAY SADD2
  @ 13, 11  SAY SADD3
  @ 14,  1  SAY "Town: "
  @ 14, 11  SAY STOWN
  @ 15,  1  SAY "Postcode: "
  @ 15, 11  SAY SPCODE
  @ 17,  1  SAY "Phone: "
  @ 17, 11  SAY SPHONE
  @ 19, 1  SAY "Current Balance: "
  @ 19, 18 SAY SPACE(11)  && blank out brackets
  @ 19,18  SAY  SBALANCE FUNCTION '(BZ'
  @ 20,1    SAY 'Credit Limit: '
  @ 20,18   SAY SCREDLIM FUNCTION 'BZ'
  @ 21,1    SAY 'Purch. to date: '
  @ 21,18   SAY PTDATE FUNCTION 'BZ'
  @  7,  0  TO 22, 37

  IF DELETED()
   @ 6,1 SAY 'Record marked for deletion'
  ENDIF

 ENDIF


 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(locsline,sel*2-1,2))
 hilite=SUBSTR(sline,VAL(SUBSTR(locsline,sel*2-1,2)),;
                     VAL(SUBSTR(lensline,sel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'smsg'+STR(sel,1) TO smsgnum
 @ 5,1 SAY &smsgnum

  I=0

* routine to deal with dbaseIII bug - not trapping leftarrow
```

```
X=CHR(200)
DO WHILE X=CHR(200)
 CALL INKEY WITH X
ENDDO
I=ASC(X)

@ 6,1 SAY SPACE(27)
@ 23,1 SAY SPACE(41)

DO CASE

 CASE I=28
  DO HMASTER

 CASE I=1
  sel=1

 CASE I=6
  sel=7

 CASE I=4  && right arrow
  IF sel=7
    sel=1
   ELSE
    sel=sel+1
  ENDIF

 CASE I=19  && left arrow
  IF sel=1
    sel=7
   ELSE
    sel=sel-1
  ENDIF

 CASE I=13  && return

  DO CASE

   CASE sel=1  && next
    IF .NOT. EOF()
     SKIP
    ENDIF
    IF EOF()
     @ 23,1 SAY 'End of file - no more records!'
     GO BOTTOM
    ENDIF

   CASE sel=2  && previous
    IF .NOT. BOF()
     SKIP-1
    ELSE
     @ 23,1 SAY 'Start of file - no more records!'
    ENDIF
```

```
      CASE sel=3  && find
        dofind=.T.
        DO WHILE dofind
         ename=SPACE(8)
         @ 8,13 GET ename FUNCTION '!'
         @ 9,13 SAY SPACE (20)
         @ 11,11 SAY SPACE(20)
         @ 12,11 SAY SPACE(20)
         @ 13,11 SAY SPACE(20)
         @ 14,11 SAY SPACE(15)
         @ 15,11 SAY SPACE (8)
         @ 17,11 SAY SPACE(15)
         @ 19,18 SAY SPACE(8)
         @ 20,18 SAY SPACE(8)
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HSMAS
          LOOP
         ENDIF
         IF ename=SPACE(8)
          dofind=.F.
          LOOP
         ENDIF
         ename=RTRIM(ename)
         SEEK ename
         IF .NOT. FOUND()
           @  23,1 SAY 'Supplier does not exist.  Moved to start of
file'
           GO TOP
          ELSE
           @  23,1 SAY 'Supplier found - select  NEXT  for  others.
'
         ENDIF
         dofind=.F.
        ENDDO   && dofind

      CASE sel=4  && append
        SET ORDER TO  && remove index to obtain last suppcode
        adding =.T.
        pointer=RECNO()  && store current position
        DO WHILE adding
         ename=SPACE(8)
         @ 8,13 GET ename FUNCTION '!'
         @ 9,13 SAY SPACE(20)
         @ 11,11 SAY SPACE(20)
         @ 12,11 SAY SPACE(20)
         @ 13,11 SAY SPACE(20)
         @ 14,11 SAY SPACE(15)
         @ 15,11 SAY SPACE(8)
         @ 17,11 SAY SPACE(15)
         @ 19,18 SAY SPACE(8)
         @ 20,18 SAY SPACE(8)
         @ 21,18 SAY SPACE(15)
         READ
         IF READKEY()=36 .OR. READKEY()=292
```

```
         DO HSMAS
         LOOP
        ENDIF
        IF ename=SPACE(8)
         adding=.F.
         LOOP
        ENDIF
        GO BOTTOM
        code=SUPPCODE
        IF code>9000
         code=1000
        ENDIF
        APPEND BLANK
        pointer=RECNO()  && to make sure record printed
        REPLACE SUPPCODE WITH code+1  && get unique key
        REPLACE SSHORTNAME WITH ename
        nohelp=.t.
        DO WHILE nohelp
         nohelp=.f.
         @ 9,13 GET SFULLNAME FUNCTION '!'
         @ 11,11 GET SADD1  FUNCTION '!'
         @ 12,11 GET SADD2 FUNCTION '!'
         @ 13,11 GET SADD3 FUNCTION '!'
         @ 14,11 GET STOWN FUNCTION '!'
         @ 15,11 GET SPCODE FUNCTION '!'
         @ 17,11 GET SPHONE FUNCTION '!'
* BXZ = left justify, negative in brackets, zero as blank
         @ 19,18 GET SBALANCE FUNCTION '(BZ' PICTURE '99999.99'
         @ 20,18 GET SCREDLIM FUNCTION 'BZ' PICTURE '99999.99'
         @ 21,18 GET PTDATE FUNCTION 'BZ' PICTURE '9999999.99'
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HSMAS
          nohelp=.t.
          LOOP
         ENDIF
        ENDDO && nohelp
       ENDDO  && adding
       SET ORDER TO 1   && restore index
       REINDEX
       GOTO pointer  && to make sure appended record onscreen

     CASE sel=5  && change
       nohelp=.t.
       DO WHILE nohelp
        nohelp=.f.
        @ 8,13 GET SSHORTNAME FUNCTION '!'
        @ 9,13 GET SFULLNAME FUNCTION '!'
        @ 11,11 GET SADD1 FUNCTION '!'
        @ 12,11 GET SADD2 FUNCTION '!'
        @ 13,11 GET SADD3 FUNCTION '!'
        @ 14,11 GET STOWN FUNCTION '!'
        @ 15,11 GET SPCODE FUNCTION '!'
        @ 17,11 GET SPHONE FUNCTION '!'
        @ 19,18 GET SBALANCE FUNCTION '(BZ' PICTURE '99999.99'
```

```
        @ 20,18 GET SCREDLIM FUNCTION 'BZ' PICTURE '99999.99'
        @ 21,18 GET PTDATE FUNCTION 'BZ' PICTURE '9999999.99'
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HSMAS
         nohelp=.t.
         LOOP
        ENDIF
       ENDDO  && nohelp

     CASE sel=6  && delete
       DELETE
       del=.T.
       @ 23,1 SAY 'Supplier deleted. Next one displayed'
       SKIP
       IF EOF()
        @ 23,1 SAY 'Deleted. Gone to start of file       '
        GO TOP
       ENDIF

     CASE sel=7  && exit
       menuck=.F.

     ENDCASE

    OTHERWISE
     ? CHR(7)

  ENDCASE

ENDDO && menuck

IF del
  PACK
ENDIF

SELECT 10
USE
SELECT 3
USE ORDER INDEX ONUM,OCUST ALIAS ORDER
RETURN
```

```
**************************************************************
*                                                            *
*                        ORDER.PRG                           *
*                                                            *
*                             by                             *
*                                                            *
*                       Geoff Minshull                       *
*                                                            *
*                                                            *
*©copyright Geoff Minshull, 1987                             *
**************************************************************

*
*  Called from: INIT
*

*
*  This program works with orders and invoices
*

CLEAR

menuck=.T.

* selection number

sel=1

DO WHILE menuck

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='orders and invoices menu'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

 * column location of menu items in variable line

 STORE '01163454' TO locline

 * length of items in locline

 STORE '12141604' TO lenline

 STORE   'Enter  orders    Process  orders     Produce   invoice
Exit' TO line
 STORE 'Maintain order file                  ' TO msg1
 STORE 'Use new orders to update all files ' TO msg2
 STORE 'Create and view/print invoices      ' TO msg3
 STORE 'Return to previous menu             ' TO msg4
```

```
SET COLOR TO &col1/&col2
@ 4,3 SAY line
SET COLOR TO &col2/&col1
col=VAL(SUBSTR(locline,sel*2-1,2))+2
hilite=SUBSTR(line,VAL(SUBSTR(locline,sel*2-1,2)),;
                   VAL(SUBSTR(lenline,sel*2-1,2)))
@ 4,col SAY hilite
SET COLOR TO &col1/&col2
STORE 'msg' + STR(sel,1) TO msgnum
@ 5,3 SAY &msgnum
@ 6,0 CLEAR TO 23,79

 I=0

 X=CHR(200)
 DO WHILE X=CHR(200)  && routine necessary because left arrow
  CALL INKEY WITH X   && not trapped by INKEY()
 ENDDO
 I=ASC(X)

 DO CASE
  CASE I=28
   DO HORDER

  CASE I=1  && home key
   sel=1

  CASE I=6  && end key
   sel=4

  CASE I=4  && right arrow
   IF sel=4
     sel=1
    ELSE
     sel=sel+1
   ENDIF

  CASE I=19  && left arrow
   IF sel=1
     sel=4
    ELSE
     sel=sel-1
   ENDIF

  CASE I=13  && return

   DO CASE

    CASE sel=1   && enter orders
     DO ENTORD

    CASE sel=2   && process orders
* update all files
* do not process orders previously processed, but still on file
     bar='process orders'
```

```
      SET COLOR TO &col2/&col1
      @ 24,0 SAY SPACE(80)
      @ 24,40-(LEN(bar)*.5) SAY bar
      @ 24,71 SAY DATE()
      SET COLOR TO &col1/&col2
      @ 6,3 SAY 'Please wait - this may take some time...'
      notenough=.F.     &&  boolean  for insufficient QIS to  mee
order
      unmet=0           && total unmet orders
      SELECT 10
      USE SCRATCH       && scratch file for brief reports
      SELECT ORDER
      SET ORDER TO 0
      GO TOP
      tref=SPACE(5)
      tcustcode=0
      DO WHILE .NOT. EOF()
       IF .NOT. UPDATED
        tprice=0
        tcustcode=CUSTCODE
        tref=REFERENCE
        SELECT CUSTOMER
        SET ORDER TO 2
        SEEK tcustcode
        SELECT PRODUCT
        SET INDEX TO PREF
        SEEK tref
        IF ORDER->OQUANT<=QIS
         REPLACE QIS WITH QIS-ORDER->OQUANT
         tprice=SALEPRICE
         REPLACE ORDER->UPDATED WITH .T.
        ELSE
         notenough=.T.
         unmet=unmet+1
         SELECT 10
         APPEND BLANK
         REPLACE F1 WITH STR(ORDER->ONUMBER)
         REPLACE F2 WITH CUSTOMER->FULLNAME
         REPLACE F3 WITH STR(ORDER->OQUANT)
         REPLACE F4 WITH STR(PRODUCT->QIS)
         REPLACE F5 WITH PRODUCT->REFERENCE
        ENDIF  && ORDER
        SELECT ORDER
        SET ORDER TO 0
       ENDIF && UPDATED
       SKIP
      ENDDO   && eof
      @  6,3 SAY 'Processing of  orders  complete.
      IF notenough
       reply='N'
       msg='Total    orders   where   stock   is   too   low
'+LTRIM(STR(unmet))+'  Send to printer? Y/N'
       @ 7,3 SAY msg GET reply FUNCTION '!'
       READ
       SELECT 10
```

```
GO TOP
IF reply='Y'
 @ 19,3 SAY 'Press any key when printer ready.'
 READ
 @ 19,3 SAY 'Printing.....                              '
 SET DEVICE TO PRINT
 EJECT
 @ 2,40-(LEN(comname)*.5) SAY comname
 @ 2,65 SAY DATE()
 @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
 @ 5,28 SAY 'Unmet order list'
 @ 6,28 SAY '----------------'
 @ 8,1 SAY 'Order no:'
 @ 8,15 SAY 'Full name'
 @ 8,36 SAY 'Order quan.'
 @ 8,50 SAY 'QIS'
 @ 8,60 SAY 'Prod.ref.no:'
 linecount=10

 DO WHILE .NOT. EOF()
  linecount=linecount+1
  IF linecount=50
   linecount=10
   EJECT
  ENDIF
  @ linecount,5 SAY LTRIM(RTRIM(F1))
  @ linecount,15 SAY LTRIM(RTRIM(F2))
  @ linecount,36 SAY LTRIM(RTRIM(F3))
  @ linecount,50 SAY LTRIM(RTRIM(F4))
  @ linecount,60 SAY LTRIM(RTRIM(F5))
  SKIP
 ENDDO  && eof

 EJECT
 SET DEVICE TO SCREEN
 @ 19,3 SAY 'Printing complete.'

ELSE
 @ 9,3 SAY 'Order numb: '
 @ 10,3 SAY 'Full name: '
 @ 11,3 SAY 'Order quantity: '
 @ 12,3 SAY 'Quant. in stock: '
 @ 13,3 SAY 'Prod. ref. no: '

 DO WHILE .NOT. EOF()
  @ 9,20 SAY LTRIM(F1)
  @ 10,20 SAY LTRIM(F2)
  @ 11,20 SAY LTRIM(F3)
  @ 12,20 SAY LTRIM(F4)
  @ 13,20 SAY LTRIM(F5)
  @ 19,3 SAY 'Press any key to continue'
  READ
  SKIP
  IF EOF()
   @ 19,3 SAY 'No more orders.  Press any key.'
```

```
      READ
     ENDIF

    ENDDO  && eof
   ENDIF  && reply
ENDIF  && notenough

SELECT 10  && clear and close scratch file
ZAP
USE


CASE sel=3   && produce invoices
 CLEAR
 @ 1,40-(LEN(comname)*.5) SAY comname
 @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
 bar='produce invoices'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
 RESTORE FROM INV ADDITIVE  && get last invoice number
 IF invoice=9000
  invoice=1
 ENDIF
 SELECT 10
 USE PRODHIST INDEX HGROUP
 SELECT CUSTOMER
 SET ORDER TO 2
 SELECT PRODUCT
 SET INDEX TO PREF
 SELECT ORDER
 SET ORDER TO 2
 REINDEX
 choice=' '

 DO WHILE choice<>'X'
 GO TOP
 invtot=0
 firstpass=.T.
 tempcust=0   && use to test for change in customer
 linecount=0
 choice=' '
 @ 5,1 SAY 'You can Print the invoices - P'
 @ 6,1 SAY 'or view them on the Screen - S'
 @ 7,1 SAY '                    or eXit - X'
 DO WHILE choice<>'P' .AND. choice<>'S' .AND. choice<>'X'
  @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
 ENDDO

 DO CASE

 CASE choice='P'
```

```
        @ 19,3 SAY 'Press any key when printer ready.'
        READ
        @ 19,3 SAY 'Printing....                          '
        SELECT ORDER
        SET DEVICE TO PRINT
        EJECT
       DO WHILE .NOT. EOF()
        IF UPDATED .AND. (.NOT. INVOICED)
         tcustcode=CUSTCODE
         tref=REFERENCE
         IF tempcust<>CUSTCODE
          IF .NOT. firstpass
           IF linecount=44
            EJECT
            linecount=10
           ENDIF
           @ linecount+2,48 SAY 'SUB-TOT:'
           @ linecount+2,65 SAY invtot PICTURE '9999999999.99'
           @ linecount+3,48 SAY 'VAT: '
           @     linecount+3,65    SAY    invtot*15/100    PICTURE
'9999999999.99'
           @ linecount+4,65 SAY '-------------'
           @ linecount+5,48 SAY 'TOTAL: '
           @  linecount+5,65  SAY  invtot+(invtot*15/100)  PICTURE
'9999999999.99'
           REPLACE      CUSTOMER->BALANCE      WITH      CUSTOMER-
>BALANCE+(invtot+(invtot*15/100))
           REPLACE      CUSTOMER->STDATE      WITH      CUSTOMER-
>STDATE+(invtot+(invtot*15/100))
           invtot=0
          ENDIF  && not firstpass
          firstpass=.F.
          tempcust=CUSTCODE
          invoice=invoice+1
          SELECT CUSTOMER
          SEEK tcustcode
          @ 6,1 SAY DATE()
          @ 6,40-(LEN(comname)*.5) SAY comname
          @ 6,60 SAY 'Invoice no: '
          @ 6,72 SAY LTRIM(STR(invoice))
          @ 7,40-(LEN(comadd1)*.5) SAY comadd1
          @ 8,40-(LEN(comtown)*.5) SAY comtown
          ctext='Phone: '+ RTRIM(comphone)
          @ 9,40-(LEN(ctext)*.5) SAY ctext   && to centre properly
          ctext='VAT No: '+ RTRIM(comvat)
          @ 10,40-(LEN(ctext)*.5) SAY ctext
          @ 12,1 SAY 'Invoice to: '
          @ 12,13 SAY CUSTOMER->FULLNAME
          @ 13,13 SAY TOWN
          @ 14,13 SAY 'Our ref: '
          @ 14,22 SAY CUSTCODE
          @   16,1   SAY  'Order  date    Prod.ref.    Description
Quantity    Price       Total'
          @                          17,1                          SAY
'-------------------------------------------------------------------
```

```
-------------'
         linecount=19
        ENDIF  && tempcust<>custcode

        SELECT PRODUCT
        SEEK tref
*
* first update product history file
*
        SELECT 10
        APPEND BLANK
        REPLACE PRODHIST->GROUP WITH PRODUCT->GROUP
        REPLACE PRODHIST->SALEPRICE WITH PRODUCT->SALEPRICE
        REPLACE PRODHIST->SALES WITH ORDER->OQUANT
        SELECT PRODUCT
*
*  now print invoice
*
        @ linecount,1 SAY ORDER->ODATE
        @ linecount,13 SAY PRODUCT->REFERENCE
        @ linecount,24 SAY PRODUCT->DESCRIP
        @ linecount,45 SAY ORDER->OQUANT
        @ linecount,57 SAY PRODUCT->SALEPRICE
        @ linecount,65 SAY ORDER->OQUANT*PRODUCT->SALEPRICE
        linecount=linecount+1
        IF linecount=50
         EJECT
         linecount=5
        ENDIF
        invtot=invtot+ORDER->OQUANT*PRODUCT->SALEPRICE
        REPLACE ORDER->INVOICED WITH .T.
        REPLACE ORDER->INVNUM WITH invoice
        REPLACE ORDER->INVDATE WITH DATE()
        REPLACE   ORDER->ORDTOT   WITH    (ORDER->OQUANT*PRODUCT-
>SALEPRICE)*1.15
        SELECT ORDER
       ENDIF  && UPDATED
       SKIP
       ENDDO   && eof
* deal with last record
       IF linecount=44
        EJECT
        linecount=10
       ENDIF
       IF .NOT. firstpass  && for when NO invoices
        @ linecount+2,48 SAY 'SUB-TOT:'
        @ linecount+2,65 SAY invtot PICTURE '9999999999.99'
        @ linecount+3,48 SAY 'VAT: '
        @ linecount+3,65 SAY invtot*15/100 PICTURE '9999999999.99'
        @ linecount+4,65 SAY '--------------'
        @ linecount+5,48 SAY 'TOTAL: '
        @   linecount+5,65   SAY  invtot+(invtot*15/100)   PICTURE
'9999999999.99'
        SELECT CUSTOMER
        REPLACE       CUSTOMER->BALANCE       WITH       CUSTOMER-
```

```
>BALANCE+(invtot+(invtot*15/100))
        REPLACE CUSTOMER->STDATE WITH CUSTOMER->STDATE+invtot
        EJECT
        ELSE
         SET DEVICE TO SCREEN
         @ 19,3 SAY 'NO INVOICES READY! PROCESS SOME ORDERS.'
         READ
         @ 19,3 SAY '                                                '
       ENDIF   && firstpass
       SAVE TO INV ALL LIKE invoice   && save invoice number
       SET DEVICE TO SCREEN
       IF .NOT. firstpass
        @ 19,3 SAY 'Printing complete.     '
       ENDIF

       CASE choice='S'    && screen
       SELECT ORDER
       DO WHILE .NOT. EOF()
        IF UPDATED .AND. (.NOT. INVOICED)
         tcustcode=CUSTCODE
         tref=REFERENCE
         IF tempcust<>CUSTCODE
          IF .NOT. firstpass
           IF linecount>18
            READ
            @ 13,1 CLEAR TO 23,78
            linecount=17
           ENDIF  && linecount
           @ linecount+2,48 SAY 'SUB-TOT:'
           @ linecount+2,65 SAY invtot PICTURE '9999999999.99'
           @ linecount+3,48 SAY 'VAT: '
           @    linecount+3,65    SAY    invtot*15/100     PICTURE
'9999999999.99'
           @ linecount+4,65 SAY '-------------'
           @ linecount+5,48 SAY 'TOTAL: '
           @  linecount+5,65  SAY  invtot+(invtot*15/100)  PICTURE
'9999999999.99'
           invtot=0
           READ  && used instead of WAIT
          ENDIF  && not firstpass
          firstpass=.F.
          tempcust=CUSTCODE
          SELECT CUSTOMER
          SEEK tcustcode
          @ 6,0 CLEAR TO 23,79
          @ 6,0 TO 23,79
          @ 7,1 SAY DATE()
          @ 7,20 SAY comname
          @ 7,60 SAY 'INVOICE'
          @ 9,1 SAY 'Invoice to: '
          @ 9,13 SAY CUSTOMER->FULLNAME
          @   11,1   SAY  'Order  date    Prod.ref.    Descriptio
Quantity    Price       Total'
          @                          12,1                          SA
'----------------------------------------------------------------
```

```
-------------'
          linecount=14
         ENDIF  && tempcust<>custcode

         SELECT PRODUCT
         SEEK tref
         @ linecount,1 SAY ORDER->ODATE
         @ linecount,13 SAY PRODUCT->REFERENCE
         @ linecount,24 SAY PRODUCT->DESCRIP
         @ linecount,45 SAY ORDER->OQUANT
         @ linecount,57 SAY PRODUCT->SALEPRICE
         @ linecount,65 SAY ORDER->OQUANT*PRODUCT->SALEPRICE
         linecount=linecount+1
         IF linecount=23
          READ
          @ 13,1 CLEAR TO 23,78
          linecount=14
         ENDIF
         invtot=invtot+ORDER->OQUANT*PRODUCT->SALEPRICE
         SELECT ORDER
       ENDIF  && UPDATED
       SKIP
       ENDDO   && eof
* deal with last record
       IF linecount>18
        READ
        @ 13,1 CLEAR TO 23,78
        linecount=17
       ENDIF
       IF .NOT. firstpass  && where no invoices
        @ linecount+2,48 SAY 'SUB-TOT:'
        @ linecount+2,65 SAY invtot PICTURE '9999999999.99'
        @ linecount+3,48 SAY 'VAT: '
        @ linecount+3,65 SAY invtot*15/100 PICTURE '9999999999.99'
        @ linecount+4,65 SAY '-------------'
        @   linecount+5,65  SAY   invtot+(invtot*15/100)   PICTURE
'9999999999.99'
        READ
        @ 6,0 CLEAR TO 23,79
        ELSE
         @ 19,3 SAY 'NO INVOICES READY! PROCESS SOME ORDERS.'
         READ
       ENDIF && firrstpass

       CASE choice='X'
        LOOP
       ENDCASE
      ENDDO && choice<>'x'
      CLEAR

      CASE sel=4   && exit
       menuck=.F.
     ENDCASE
*
```

```
   OTHERWISE
    ? CHR(7)
  ENDCASE
 ENDDO

menuck=.T.
sel=2

SELECT PRODUCT
SET INDEX TO

RETURN
```

```
*****************************************************************
*                                                               *
*                          CPAY.PRG                             *
*                                                               *
*                              by                               *
*                                                               *
*                        Geoff Minshull                         *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called from: INIT
*

*
*  This program processes customer payments
*

@ 6,0 CLEAR TO 23,79
bar='customer payments'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

SELECT CUSTOMER
SET ORDER TO 1
SELECT 10
USE PAYMENT INDEX PCODE

looping=.T.  &&  main control loop for finding cust
@ 7,1 TO 20,42    && draw box

DO WHILE looping
 SELECT CUSTOMER
 cfind=.T.    &&  loop for finding customer

 DO WHILE cfind
  ename=SPACE(8)
  @ 9,15 SAY SPACE(8)
  @ 10,15 SAY SPACE(20)
  @ 11,15 SAY SPACE(20)
  @ 13,20 SAY SPACE(9)
  @ 14,23 SAY SPACE(6)
  @ 9,3 SAY 'Short name: ' GET ename FUNCTION '!'
  READ

  IF READKEY()=36 .OR. READKEY()=292
   DO HPAYMEN  && help screens
   @ 7,1 TO 20,42
   LOOP
```

```
   ENDIF

   IF ename=' '
    cfind=.F.
    looping=.F.
    LOOP
   ENDIF

   @ 10,3  SAY 'Full name: '
   @ 11,3 SAY 'Address: '
   ename=RTRIM(ename)
   SEEK ename
   IF .NOT. FOUND()
    @ 19,3 SAY 'Customer does not exist.                    '
    LOOP
   ENDIF
   next=.T.  && loop to confirm correct name

   DO WHILE next
    @ 9,15 SAY SHORTNAME
    @ 10,15  SAY FULLNAME
    @ 11,15 SAY ADD1
    entry=' '

    DO WHILE entry=' '
     @  19,3   SAY 'Correct name?  Y/N or X to quit:  ' GET  entry
FUNCTION '!'
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPAYMEN
      @ 7,1 TO 20,42
      LOOP
     ENDIF
     @ 19,3 SAY SPACE(39)
    ENDDO && entry
    IF entry='X'
     next=.F.
     LOOP
    ENDIF
    IF entry='Y'
     pay=0
     number=0
     @ 13,3 SAY 'Enter payment: ' GET pay PICTURE '99999.99'
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPAYMEN
      @ 7,1 TO 20,42
      LOOP
     ENDIF
     IF pay=0
      cfind=.F.
      looping=.F.
      LOOP
     ENDIF
     @ 14,3 SAY 'Enter invoice no: ' GET number PICTURE '99999'
```

```
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPAYMEN
      @ 7,1 TO 20,42
      LOOP
     ENDIF
     SELECT 10
     APPEND BLANK
     REPLACE AMOUNT WITH pay
     REPLACE CUSTCODE WITH CUSTOMER->CUSTCODE
     REPLACE PDATE WITH DATE()
     REPLACE INVNUM WITH number
     REPLACE CUSTOMER->BALANCE WITH CUSTOMER->BALANCE-pay
     next=.F.
     cfind=.F.
     LOOP
    ENDIF
    SKIP
    IF EOF()
     @ 19,3 SAY 'No more records. Press any key to quit.'
     READ
     @ 19,3 SAY SPACE(39)
     next=.F.
     LOOP

   ENDIF  && entry=y
  ENDDO &&next
 ENDDO  && cfind
ENDDO  && looping
*
*  close file
*
SELECT 10
USE
RETURN
```

```
*************************************************************
*                                                           *
*                        SPAY.PRG                           *
*                                                           *
*                           by                              *
*                                                           *
*                      Geoff Minshull                       *
*                                                           *
*                                                           *
*©copyright Geoff Minshull, 1987                            *
*************************************************************

*
*  Called from: INIT
*

*
*  This program processes supplier payments. In fact, all it
* does is reduce the balance in the Supplier file. No record
* is kept of the payments, and they are not audit trailed.


@ 6,0 CLEAR TO 23,79
bar='supplier payments'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

SELECT 10
USE SUPPLIER INDEX SNAME

looping=.T.   &&  main control loop for finding supplier
@ 7,1 TO 20,42    && draw box

DO WHILE looping
 cfind=.T.     &&  loop for finding supplier
 DO WHILE cfind
  ename=SPACE(8)
  @ 9,15 SAY SPACE(8)
  @ 10,15 SAY SPACE(20)
  @ 11,15 SAY SPACE(20)
  @ 13,20 SAY SPACE(9)
  @ 14,23 SAY SPACE(6)
  @ 9,3 SAY 'Short name: ' GET ename FUNCTION '!'
  READ
  IF READKEY()=36 .OR. READKEY()=292
   DO HPAYMEN
   @ 7,1 TO 20,42
   LOOP
  ENDIF
  IF ename=' '
   cfind=.F.
```

```
    looping=.F.
    LOOP
   ENDIF
   @ 10,3  SAY 'Full name: '
   @ 11,3 SAY 'Address: '
   ename=RTRIM(ename)
   SEEK ename
   IF .NOT. FOUND()
    @ 19,3 SAY 'Supplier does not exist.                    '
    LOOP
   ENDIF
   next=.T.  && loop to confirm correct name
   DO WHILE next
    @ 9,15 SAY SSHORTNAME
    @ 10,15  SAY SFULLNAME
    @ 11,15 SAY SADD1
    entry=' '
    DO WHILE entry=' '
     @  19,3   SAY 'Correct name?  Y/N or X to quit:  ' GET  entry
FUNCTION '!'
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPAYMEN
      @ 7,1 TO 20,42
      LOOP
     ENDIF
     @ 19,3 SAY SPACE(39)
    ENDDO && entry
    IF entry='X'
     next=.F.
     LOOP
    ENDIF
    IF entry='Y'
     pay=0
     number=0
     @ 13,3 SAY 'Enter payment: ' GET pay PICTURE '99999.99'
     READ
     IF READKEY()=36 .OR. READKEY()=292
      DO HPAYMEN
      @ 7,1 TO 20,42
      LOOP
     ENDIF
     IF pay=0
      cfind=.F.
      looping=.F.
      LOOP
     ENDIF
     REPLACE SBALANCE WITH SBALANCE-pay
     next=.F.
     cfind=.F.
     LOOP
    ENDIF
    SKIP
    IF EOF()
     @ 19,3 SAY 'No more records. Press any key to quit.'
```

```
      READ
      @ 19,3 SAY SPACE(39)
      next=.F.
      LOOP
     ENDIF  && entry=y
    ENDDO &&next
   ENDDO  && cfind
ENDDO  && looping
*
*  close file
*
USE
RETURN
```

```
*************************************************************
*                                                           *
*                       REPORTS.PRG                         *
*                                                           *
*                            by                             *
*                                                           *
*                      Geoff Minshull                       *
*                                                           *
*                                                           *
*(c)copyright Geoff Minshull, 1987                          *
*************************************************************

*
*  Called from: INIT
*

*
*  This program is the report menu, and some routines.
*


CLEAR

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='reports menu'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

rmenuck=.T.
rsel=1
lstrsel=19  && number of choices
I=0

STORE 'Customers: list of those over credit limit' TO msg1
STORE 'Customers: letter to those over credit limit' TO msg2
STORE 'Customers: statements' TO msg3
STORE 'Customers: list all customers to printer' TO msg4
STORE ' ' TO msg5
STORE 'Products: reorder list' TO msg6
STORE 'Products: order to supplier' TO msg7
STORE 'Products: stock valuation by group to printer' TO msg8
STORE 'Products: stock over maximum list' TO msg9
STORE 'Products: list all products to printer' TO msg10
STORE 'Products: produce bar charts ' TO msg11
STORE ' ' TO msg12
STORE 'Suppliers: list all suppliers to printer' TO msg13
STORE ' ' TO msg14
STORE 'Sales ledger' TO msg15
STORE 'Audit trail to printer' TO msg16
STORE 'Address labels' TO msg17
```

```
STORE ' ' TO msg18
STORE 'Return to previous menu' TO msg19

SET COLOR TO &col2/&col1
@ 4,1 SAY msg1
SET COLOR TO &col1/&col2
@ 5,1 SAY msg2
@ 6,1 SAY msg3
@ 7,1 SAY msg4
@ 8,1 SAY msg5
@ 9,1 SAY msg6
@ 10,1 SAY msg7
@ 11,1 SAY msg8
@ 12,1 SAY msg9
@ 13,1 SAY msg10
@ 14,1 SAY msg11
@ 15,1 SAY msg12
@ 16,1 SAY msg13
@ 17,1 SAY msg14
@ 18,1 SAY msg15
@ 19,1 SAY msg16
@ 20,1 SAY msg17
@ 21,1 SAY msg18
@ 22,1 SAY msg19



DO WHILE rmenuck

IF I=13  && redraw menu
 CLEAR
 @ 1,40-(LEN(comname)*.5) SAY comname
 @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
 @ 4,1 SAY msg1
 @ 5,1 SAY msg2
 @ 6,1 SAY msg3
 @ 7,1 SAY msg4
 @ 8,1 SAY msg5
 @ 9,1 SAY msg6
 @ 10,1 SAY msg7
 @ 11,1 SAY msg8
 @ 12,1 SAY msg9
 @ 13,1 SAY msg10
 @ 14,1 SAY msg11
 @ 15,1 SAY msg12
 @ 16,1 SAY msg13
 @ 17,1 SAY msg14
 @ 18,1 SAY msg15
 @ 19,1 SAY msg16
 @ 20,1 SAY msg17
 @ 21,1 SAY msg18
 @ 22,1 SAY msg19
 bar='reports menu'
 SET COLOR TO &col2/&col1
```

```
 @ 24,0 SAY SPACE(80)
 @ 24,1 SAY '<F1> for help'
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
ENDIF  && I=13

 STORE 'msg' + LTRIM(STR(lstrsel,2)) TO msgnum
 @ lstrsel+3,1 SAY &msgnum  && numbers may change with spacing
 STORE 'msg' + LTRIM(STR(rsel,2)) TO msgnum
 SET COLOR TO &col2/&col1
 @ rsel+3,1 SAY &msgnum
 SET COLOR TO &col1/&col2



  I=0
  DO WHILE I=0
   I=INKEY()   && inkey ok here because leftarrow not used
  ENDDO

  lstrsel=rsel

  DO CASE

   CASE I=28
    DO HREPORTS

   CASE I=1
    rsel=1

   CASE I=6
    rsel=19

   CASE I=24  && down arrow
   IF rsel=19
     rsel=1
    ELSE
     rsel=rsel+1
   ENDIF
   IF  rsel=5 .OR.  rsel=12 .OR.  rsel=14 .OR.  rsel=18  &&  skip
blanks
    rsel=rsel+1
   ENDIF

   CASE I=5  && up arrow
   IF rsel=1
     rsel=19
    ELSE
     rsel=rsel-1
   ENDIF
   IF  rsel=5 .OR.  rsel=12 .OR.  rsel=14 .OR.  rsel=18  &&  skip
blanks
    rsel=rsel-1
   ENDIF
```

```
    CASE I=13

     DO CASE

      CASE rsel=1   && over credit list
       CLEAR
       bar='customers over credit limit'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
      choice=' '
       DO WHILE choice<>'X'
       SELECT CUSTOMER
       SET ORDER TO 1
       GO TOP
       choice=' '
       @ 3,1 SAY 'You can get a Screen report  -  S'
       @ 4,1 SAY '         or a  Printed report -  P'
       @ 5,1 SAY '         or eXit              -  X'
       DO WHILE choice<>'S' .AND. choice<>'P' .AND. choice<>'X'
        @ 3,40 SAY 'Enter your choice: ' GET choice FUNCTION '!'
        READ
       ENDDO

       DO CASE
        CASE choice='S'
         tot=0
         linecount=8
         @ 8,0 CLEAR TO 23,79
         @   8,0   SAY  'Shortname   Town                          Phone
Limit     Balance     Excess'
         DO WHILE .NOT. EOF()
          IF BALANCE>CREDLIM
           linecount=linecount+1
           IF linecount=22
            @ 23,1 SAY 'Press any key....'
            READ
            linecount=9
            @ 8,0 CLEAR TO 23,79
           ENDIF && linecount
           @ linecount,0 SAY SHORTNAME
           @ linecount,11 SAY TOWN
           @ linecount,33 SAY PHONE
           @ linecount,49 SAY CREDLIM
           @ linecount,58 SAY BALANCE
           @ linecount,69 SAY BALANCE-CREDLIM
           tot=tot+BALANCE-CREDLIM
          ENDIF  && balance
          SKIP
         ENDDO  eof
         @ 23,30 SAY 'Total excess = '
```

```
          @ 23,43 SAY tot
          READ

         CASE choice='P'
          tot=0
          linecount=8
          @ 19,3 SAY 'Press any key when printer ready.'
          READ
          @ 9,0 CLEAR TO 23,79
          @ 19,3 SAY 'Printing.....'
          SET DEVICE TO PRINT
          EJECT
          @ 2,40-(LEN(comname)*.5) SAY comname
          @ 2,65 SAY DATE()
          @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
          @ 5,26 SAY 'Customers over credit limit'
          @ 6,26 SAY '----------------------------'
          @ linecount,0 SAY 'Shortname  Town                          Phone
Limit     Balance    Excess'
          DO WHILE .NOT. EOF()
           IF BALANCE>CREDLIM
            linecount=linecount+1
            IF linecount=50
             EJECT
             linecount=10
            ENDIF && linecount
            @ linecount,0 SAY SHORTNAME
            @ linecount,11 SAY TOWN
            @ linecount,33 SAY PHONE
            @ linecount,49 SAY CREDLIM
            @ linecount,58 SAY BALANCE
            @ linecount,69 SAY BALANCE-CREDLIM
            tot=tot+BALANCE-CREDLIM
           ENDIF  && balance
           SKIP
          ENDDO  eof
          @ linecount+4,30 SAY 'Total excess = '
          @ linecount+4,43 SAY tot
          EJECT
          SET DEVICE TO SCREEN
          @ 19,3 SAY 'Printing complete.    '

         CASE choice='X'
          LOOP

         ENDCASE

         ENDDO   && choice<>'X'

        CASE rsel=2  && over credit letter
         CLEAR
         bar='letter to customers over credit limit'
         SET COLOR TO &col2/&col1
         @ 24,0 SAY SPACE(80)
         @ 24,40-(LEN(bar)*.5) SAY bar
```

```
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2
SELECT CUSTOMER
SET ORDER TO 1
choice=' '
DO WHILE choice<>'X'
 GO TOP
 choice=' '
 @ 5,1 SAY 'You can Print the letter - P'
 @ 6,1 SAY '                    or eXit - X'
 DO WHILE choice<>'P' .AND. choice<>'X'
  @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
 ENDDO

 DO CASE
  CASE choice='P'
   @ 19,3 SAY 'Press any key when printer ready.'
   READ
   @ 9,0 CLEAR TO 23,79
   @ 19,3 SAY 'Printing.....'
   SET DEVICE TO PRINT
   DO WHILE .NOT. EOF()
    IF BALANCE>CREDLIM
     EJECT
     @ 4,35 SAY 'REMINDER'
     @ 4,60 SAY DATE()
     @ 7,8 SAY comname
     @ 7,50 SAY 'TO: '
     @ 7,55 SAY FULLNAME
     @ 8,8 SAY comadd1
     @ 8,55 SAY ADD1
     @ 9,8 SAY comtown
     @ 9,55 SAY ADD2
     @ 10,8 SAY compcode
     @ 10,55 SAY ADD3
     @ 11,55 SAY TOWN
     @ 12,55 SAY PCODE
     @ 15,8 SAY 'Dear Sir,'
     @ 17,8 SAY 'Our records show that your current balance
is now '
     @ 17,57 SAY BALANCE-CREDLIM
     @ 18,8 SAY 'over your credit limit of '
     @ 18,34 SAY CREDLIM
     @ 20,8 SAY 'We would appreciate a payment from you  as
soon as possible,'
     @ 22,8 SAY 'Yours faithfully,'
     @ 28,8 SAY 'R. SMITH'
     @ 29,8 SAY '(credit control)'
    ENDIF  && balance
    SKIP
   ENDDO  eof
   EJECT
   SET DEVICE TO SCREEN
   @ 19,3 SAY 'Printing complete.    '
```

```
        CASE choice='X'
           LOOP
          ENDCASE
         ENDDO  && choice<>x

      CASE rsel=3  && statement
       bar='customer statements'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,1 SAY '<F1> for help'
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
       DO STAT

      CASE rsel=4  && customer list
       CLEAR
       bar='print all customers'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
       SELECT CUSTOMER
       SET ORDER TO 1
       choice=' '
       DO WHILE choice<>'X'
        GO TOP
        linecount=8
        choice=' '
        @ 5,1 SAY 'You can Print the report - P'
        @ 6,1 SAY '                    or eXit - X'
        DO WHILE choice<>'P' .AND. choice<>'X'
         @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
         READ
        ENDDO

       DO CASE
        CASE choice='P'
        @ 19,3 SAY 'Press any key when printer ready.'
        READ
        @ 9,0 CLEAR TO 23,79
        @ 19,3 SAY 'Printing.....'
        SET DEVICE TO PRINT
        EJECT
        @ 2,40-(LEN(comname)*.5) SAY comname
        @ 2,65 SAY DATE()
        @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
        @ 5,26 SAY 'List of all customers'
        @ 6,26 SAY '----------------------'
        @   linecount,0  SAY  'Code   Fullname                    Town
Limit     Balance   Sales ex. VAT'

       DO WHILE .NOT. EOF()
```

```
         linecount=linecount+1
         IF linecount=50
          EJECT
          @    8,0    SAY   'Code     Fullname                        Town
Limit     Balance   Sales ex. VAT'
          linecount=10
         ENDIF
         @ linecount,0 SAY CUSTCODE
         @ linecount,6 SAY FULLNAME
         @ linecount,28 SAY TOWN
         @ linecount,45 SAY CREDLIM
         @ linecount,54 SAY BALANCE
         @ linecount,64 SAY STDATE
         SKIP
        ENDDO  && eof
        EJECT
        SET DEVICE TO SCREEN
        @ 19,3 SAY 'Printing complete.    '

        CASE choice='X'
         LOOP

        ENDCASE

       ENDDO  && choice<>x


       CASE rsel=6   && reorder list
        bar='reorder list'
        SET COLOR TO &col2/&col1
        @ 24,0 SAY SPACE(80)
        @ 24,1 SAY '<F1> for help'
        @ 24,40-(LEN(bar)*.5) SAY bar
        @ 24,71 SAY DATE()
        SET COLOR TO &col1/&col2
        MAX=.F.
        DO REOMAX

       CASE rsel=7  &&  order to supplier
        CLEAR
        bar='print orders to suppliers'
        SET COLOR TO &col2/&col1
        @ 24,0 SAY SPACE(80)
        @ 24,40-(LEN(bar)*.5) SAY bar
        @ 24,71 SAY DATE()
        SET COLOR TO &col1/&col2
        SELECT 10
        USE SUPPLIER INDEX SCODE
        GO TOP
        SELECT PRODUCT
        SET INDEX TO PSUPP
        REINDEX
        choice=' '
        DO WHILE choice<>'X'
         GO TOP
```

```
        choice=' '
        @ 5,1 SAY 'You can Print the orders - P'
        @ 6,1 SAY '                    or eXit - X'
        DO WHILE choice<>'P' .AND. choice<>'X'
         @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
         READ
        ENDDO

       DO CASE

        CASE choice='P'
        @ 19,3 SAY 'Press any key when printer ready.'
        READ
        @ 9,0 CLEAR TO 23,79
        @ 19,3 SAY 'Printing.....'
*        SET DEVICE TO PRINT
        repeat=0  && control for >1 order from same supplier
        linecount=0
        DO WHILE .NOT. EOF()
         IF QIS<MINSTOCK
         key=SUPPCODE
         SELECT 10
         SEEK key
         IF .NOT. FOUND()
          EJECT
          @ 10,5 SAY 'Supplier code '
          @ 10,19 SAY PRODUCT->SUPPCODE
          @  10,24 SAY 'does not exist.  Please check the code  in
Product file.'
          @ 12,5 SAY 'Product code is '
          @ 12,21 SAY PRODUCT->REFERENCE
         ELSE
          IF repeat<>PRODUCT->SUPPCODE
           @ 4,35 SAY 'ORDER'
           @ 4,60 SAY DATE()
           @ 7,8 SAY comname
           @ 7,50 SAY 'TO: '
           @ 7,55 SAY SFULLNAME
           @ 8,8 SAY comadd1
           @ 8,55 SAY SADD1
           @ 9,8 SAY comtown
           @ 9,55 SAY SADD2
           @ 10,8 SAY compcode
           @ 10,55 SAY SADD3
           @ 11,8 SAY 'Phone: '
           @ 11,15 SAY comphone
           @ 11,55 SAY STOWN
           @ 12,8 SAY 'VAT No: '
           @ 12,16 SAY comvat
           @ 12,55 SAY SPCODE
           @ 15,8 SAY 'Dear Sir,'
           @  17,8  SAY 'We would like to re-order  the  following
item(s):'
           @      19,10      SAY      'Code              Description
Quant. required'
```

```
      linecount=20
     ENDIF && repeat
     linecount=linecount+1
     IF linecount=50
      EJECT
      linecount=10
     ENDIF
     @ linecount,10 SAY PRODUCT->REFERENCE
     @ linecount,21 SAY PRODUCT->DESCRIP
     @ linecount,51 SAY PRODUCT->ROQ
    ENDIF  && FOUND()
   ENDIF  && qis
   SELECT PRODUCT
   repeat=PRODUCT->SUPPCODE
   SKIP
  ENDDO  && eof
  EJECT
  SELECT PRODUCT
  SET INDEX TO
  SET DEVICE TO SCREEN
  @ 19,3 SAY 'Printing complete.    '

CASE choice='X'
 SELECT 10
 USE
 LOOP

ENDCASE

ENDDO  && choice<>x

CASE rsel=8  && stock valuation
 CLEAR
 @ 1,40-(LEN(comname)*.5) SAY comname
 @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
 bar='print stock valuation'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
 SELECT PRODUCT
 SET INDEX TO PGROUP
 linecount=9
 choice=' '
 DO WHILE choice<>'X'
  GO TOP
  choice=' '
  @ 5,1 SAY 'You can Print the report - P'
  @ 6,1 SAY '                    or eXit - X'
  DO WHILE choice<>'P' .AND. choice<>'X'
   @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
   READ
  ENDDO
```

```
        DO CASE
         CASE choice='P'
          @ 19,3 SAY 'Press any key when printer ready.'
          READ
          @ 9,0 CLEAR TO 23,79
          @ 19,3 SAY 'Printing.....'
          SET DEVICE TO PRINT
          EJECT
          @ 2,40-(LEN(comname)*.5) SAY comname
          @ 2,65 SAY DATE()
          @     3,40-(LEN(comname)*.5)     SAY     REPLICATE      ('-
',LEN(comname))
          @ 5,28 SAY 'Stock valuation by group'
          @ 6,28 SAY '------------------------'
          @  8,0  SAY 'Code   Desc.                        Group      QIS
Value (cost)         Value (sale)'
          tgroup=GROUP
          gcost=0
          gsale=0
          totcost=0
          totsale=0
          DO WHILE .NOT. EOF()
          linecount=linecount+1
          IF linecount=50
           EJECT
           @  8,0  SAY 'Code  Desc.                        Group      QIS
Value (cost)         Value (sale)'
           linecount=10
          ENDIF
          IF GROUP<>tgroup
           @ linecount+1,30 SAY 'Group totals: '
           @ linecount+1,44 SAY gcost
           @ linecount+1,64 SAY gsale
           gcost=0
           gsale=0
           linecount=linecount+4
          ENDIF
          @ linecount,0 SAY REFERENCE
          @ linecount,6 SAY DESCRIP
          @ linecount,28 SAY GROUP
          @ linecount,35 SAY QIS
          @ linecount,44 SAY QIS*COSTPRICE
          @ linecount,64 SAY QIS*SALEPRICE
          gcost=gcost+QIS*COSTPRICE
          gsale=gsale+QIS*SALEPRICE
          totcost=totcost+QIS*COSTPRICE
          totsale=totsale+QIS*SALEPRICE
          tgroup=GROUP
          SKIP
         ENDDO  && eof
         @ linecount+2,30 SAY 'Group totals: '
         @ linecount+2,44 SAY gcost
         @ linecount+2,64 SAY gsale
         @ linecount+4,30 SAY 'TOTALS: '
         @ linecount+4,44 SAY totcost
```

```
    @ linecount+4,64 SAY totsale
    EJECT
    SET DEVICE TO SCREEN
    @ 19,3 SAY 'Printing complete.    '

   CASE choice='X'
    LOOP
   ENDCASE
  ENDDO  && choice='X'

 CASE rsel=9  && stock over max
  bar='list stock over maximum'
  SET COLOR TO &col2/&col1
  @ 24,0 SAY SPACE(80)
  @ 24,1 SAY '<F1> for help'
  @ 24,40-(LEN(bar)*.5) SAY bar
  @ 24,71 SAY DATE()
  SET COLOR TO &col1/&col2
  MAX=.T.
  DO REOMAX

 CASE rsel=10  && product list
  CLEAR
  bar='print all products'
  SET COLOR TO &col2/&col1
  @ 24,0 SAY SPACE(80)
  @ 24,40-(LEN(bar)*.5) SAY bar
  @ 24,71 SAY DATE()
  SET COLOR TO &col1/&col2

  choice=' '
  DO WHILE choice<>'X'
  SELECT PRODUCT
  choice=' '
  @ 3,1 SAY 'You can get a list by Product Code  -  P'
  @ 4,1 SAY '          or a list by Group        -  G'
  @ 5,1 SAY '          or eXit                   -  X'
  DO WHILE choice<>'P' .AND. choice<>'G' .AND. choice<>'X'
   @ 3,45 SAY 'Enter your choice: ' GET choice FUNCTION '!'
   READ
  ENDDO
  IF choice='X'
   LOOP
  ENDIF
  IF choice='P'
   SET INDEX TO PREF
   ELSE
   SET INDEX TO PGROUP
  ENDIF
  GO TOP
  linecount=8
  @ 19,3 SAY 'Press any key when printer ready.'
  READ
  @ 9,0 CLEAR TO 23,79
  @ 19,3 SAY 'Printing.....'
```

```
        SET DEVICE TO PRINT
        EJECT
        @ 2,40-(LEN(comname)*.5) SAY comname
        @ 2,65 SAY DATE()
        @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
        IF choice='P'
         @ 5,26 SAY 'List of products by code'
         @ 6,26 SAY '-------------------------'
        ELSE
         @ 5,26 SAY 'List of products by group'
         @ 6,26 SAY '-------------------------'
        ENDIF
        @  7,0  SAY 'Code/desc.   Group   Cost       Sale       QIS
Min.    Max.   Reorder   Supplier'
        @    8,0    SAY   '                        price        price
stock   stock  quantity          '
        DO WHILE .NOT. EOF()
         linecount=linecount+2
         IF linecount=50
          EJECT
          @  7,0  SAY 'Code/desc.   Group  Cost       Sale       QIS
Min.    Max.   Reorder   Supplier'
          @    8,0   SAY   '                        price        price
stock   stock  quantity          '
          linecount=10
         ENDIF
         @ linecount,0 SAY REFERENCE
         @ linecount,12 SAY GROUP
         @ linecount,19 SAY COSTPRICE
         @ linecount,29 SAY SALEPRICE
         @ linecount,39 SAY QIS
         @ linecount,46 SAY MINSTOCK
         @ linecount,54 SAY MAXSTOCK
         @ linecount,61 SAY ROQ
         @ linecount,71 SAY SUPPCODE
         @ linecount+1,0  SAY DESCRIP
         SKIP
        ENDDO  && eof
        EJECT
        SET DEVICE TO SCREEN
        @ 19,3 SAY 'Printing complete.   '

      ENDDO  && choice='X'

      CASE rsel=11
       DO BARCHART

      CASE rsel=13  && supplier list
       CLEAR
       bar='print all suppliers'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
```

```
SELECT 10
USE SUPPLIER INDEX SNAME
choice=' '
DO WHILE choice<>'X'
 GO TOP
 linecount=2
 choice=' '
 @ 5,1 SAY 'You can Print the report - P'
 @ 6,1 SAY '                    or eXit - X'
 DO WHILE choice<>'P' .AND. choice<>'X'
  @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
 ENDDO

DO CASE
CASE choice='P'
@ 19,3 SAY 'Press any key when printer ready.'
READ
@ 9,0 CLEAR TO 23,79
@ 19,3 SAY 'Printing.....'
SET DEVICE TO PRINT
EJECT
@ 2,40-(LEN(comname)*.5) SAY comname
@ 2,65 SAY DATE()
@ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
@ 5,26 SAY 'List of all suppliers'
@ 6,26 SAY '----------------------'
@ 8,0 SAY 'Name/address          Code Shortname          Limit
Balance    Tot. Purchases'
DO WHILE .NOT. EOF()
 linecount=linecount+8
 IF linecount=50
  EJECT
  @   8,0   SAY   'Name/address            Code    Shortname
Limit      Balance    Tot. Purchases'
  linecount=10
 ENDIF
 @ linecount,0 SAY SFULLNAME
 @ linecount,21 SAY SUPPCODE
 @ linecount,26 SAY SSHORTNAME
 @ linecount,43 SAY SCREDLIM
 @ linecount,54 SAY SBALANCE
 @ linecount,65 SAY PTDATE
 @ linecount+1,0 SAY SADD1
 @ linecount+2,0 SAY SADD2
 @ linecount+3,0 SAY SADD3
 @ linecount+4,0 SAY STOWN
 @ linecount+5,0 SAY SPCODE
 @ linecount+6,0 SAY SPHONE
 SKIP
ENDDO  && eof
EJECT
SET DEVICE TO SCREEN
@ 19,3 SAY 'Printing complete.    '
```

```
        CASE choice='X'
         USE
         LOOP
        ENDCASE
        ENDDO  && choice<>x


       CASE rsel=15  && sales ledger
        DO LEDGER

       CASE rsel=16
        DO AUDIT

       CASE rsel=17
        DO MAILLIST

       CASE rsel=19   && exit option
        rmenuck=.F.

     ENDCASE
*
*
  OTHERWISE
   ? CHR(7)

 ENDCASE

ENDDO

CLEAR
RETURN
```

```
*****************************************************************
*                                                               *
*                        BACKUP.PRG                             *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*(c)copyright Geoff Minshull, 1987                              *
*****************************************************************

*
*  Called from: INIT
*

*
*  This program backups/restores data files. Note that after
* restoring, the file will need reindexing.
* Files must be closed before COPY command can be used.
* Note that only Customer, Product, Order and Supplier are
* backed up. Backing up via dBASE is to the SAME disk - on a
* twin floppy system, not possible to do any other way from
* within dbase. If using hard disk, data files would be copied
* to hard disk (if originally on floppy) then back to another
* floppy, using basically same routine as below.
*

CLEAR
@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE('-',LEN(comname))
bar='backup/restore files'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

choice=' '
DO WHILE choice<>'X'
 choice=' '
 @ 5,1 SAY 'You can Backup existing data files - B'
 @ 6,1 SAY '      or Restore files from backups - R'
 @ 7,1 SAY '      or eXit                       - X'

 DO WHILE choice<>'B' .AND. choice<>'R' .AND. choice<>'X'
  @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
  @ 10,1 SAY SPACE(50)
  IF READKEY()=36
   DO HBACKUP
  ENDIF
 ENDDO
```

```
 DO CASE
  CASE choice='B'
   @ 10,1 SAY 'Please be patient - this is a long job....'

* first check diskspace sufficient for backups

   totbytes=0
   SELECT CUSTOMER
   totbytes=totbytes+(RECSIZE()*RECCOUNT()+4200)
   SELECT PRODUCT
   totbytes=totbytes+(RECSIZE()*RECCOUNT()+4200)
   SELECT ORDER
   totbytes=totbytes+(RECSIZE()*RECCOUNT()+4200)
   SELECT 10
   USE SUPPLIER
   totbytes=totbytes+(RECSIZE()*RECCOUNT()+4200)
    IF DISKSPACE()<=totbytes

*  note  that  this is very crude measure - the  previous  backup
files
* should be (but aren't) ignored in this calculation since
* they are overwritten

     @ 12,1 SAY 'Sorry - not enough diskspace to make backups.'
     @ 13,1 SAY 'Press any key to exit this menu.'
     READ
     choice='X'
     USE   && close supplier
     LOOP
    ENDIF

    CLOSE ALL DATABASES   && necessary before backing up
    COPY FILE CUSTOMER.DBF TO CUSTBACK.DBF
    COPY FILE PRODUCT.DBF TO PRODBACK.DBF
    COPY FILE ORDER.DBF TO ORDBACK.DBF
    COPY FILE SUPPLIER.DBF TO SUPPBACK.DBF

*
* reopen all files
*
    SET PROCEDURE TO HELP1
    SELECT 1
    USE CUSTOMER INDEX CNAME,CCODE ALIAS CUSTOMER
    SELECT 2
    USE PRODUCT ALIAS PRODUCT
    SELECT 3
    USE ORDER INDEX ONUM,OCUST ALIAS ORDER
    @ 10,1 SAY SPACE(50)
    @ 10,1 SAY 'Backups complete. Bytes left on disk = '
    @ 10,41 SAY LTRIM(STR(DISKSPACE()))

  CASE choice='R'
   @ 10,1 SAY 'Please be patient - this is a long job....'

   CLOSE ALL DATABASES
```

```
    COPY FILE CUSTBACK.DBF TO CUSTOMER.DBF
    COPY FILE PRODBACK.DBF TO PRODUCT.DBF
    COPY FILE ORDBACK.DBF TO ORDER.DBF
    COPY FILE SUPPBACK.DBF TO SUPPLIER.DBF

    SET PROCEDURE TO HELP1
    SELECT 2
    USE PRODUCT INDEX PDESC,PREF,PGROUP,PSUPP ALIAS PRODUCT
    REINDEX
    CLOSE INDEX
    SELECT 1
    USE CUSTOMER INDEX CNAME,CCODE ALIAS CUSTOMER
    REINDEX
    SELECT 3
    USE ORDER INDEX ONUM,OCUST,ODATE ALIAS ORDER
    REINDEX
    CLOSE INDEX
    SET INDEX TO ONUM,OCUST
    @ 10,1 SAY SPACE(50)
    @ 10,1 SAY 'Restoring complete.'

  CASE choice='X'
   LOOP
 ENDCASE
ENDDO  && choice<>'x'

RETURN
```

```
*****************************************************************
*                                                               *
*                        ENTPASS.PRG                            *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*(c)copyright Geoff Minshull, 1987                              *
*****************************************************************

*
*  Called from: INIT
*

*
*  This program is not intended to provide real security -
* however, it does allow the manager the facility to examine and
* add/delete/change existing passwords. Level 1 is lowest
* access, 4 highest. Data stored in file PASS.DBF.
* Routine is devised so that staff can set up their own users
* to fit in with any exercises they may write. Staff have
* access level 4 - which allows access to this routine.
* Students would normally have only access levels 1-3. It
* is recommended that staff set up their own password as
* soon as possible.
*

CLEAR

SELECT 10
USE PASS INDEX UPASS

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))

bar='maintain users and passwords'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

menuck=.T.
del=.F.  && pack only if deletions made

* selection number

sel=1
l=13  && to make sure record printed on first pass

* column location of menu items in variable zline
```

```
STORE '01102333425466' TO loczline

* length of items in loczline

STORE '04080403060604' TO lenzline

STORE    'Next       Previous       Find        Add         Chang
Delete      Exit' TO zline
STORE 'Find next user          ' TO zmsg1
STORE 'Find previous user      ' TO zmsg2
STORE 'Find record by user     ' TO zmsg3
STORE 'Add new user            ' TO zmsg4
STORE 'Change user details     ' TO zmsg5
STORE 'Delete a user           ' TO zmsg6
STORE 'Return to previous menu' TO zmsg7

DO WHILE menuck

 @ 4,1 SAY zline

 IF I=13 .AND. (.NOT. EOF())
  @ 8,3 TO 14,25
  @ 9,5 SAY 'USER: '
  @ 9,11 SAY USER
  @ 11,5 SAY 'PASSWORD: '
  @ 11,15 SAY PASSWORD
  @ 13,5 SAY 'ACCESS LEVEL: '
  @ 13,19 SAY ACCESS FUNCTION 'BZ'
 ENDIF

 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(loczline,sel*2-1,2))
 hilite=SUBSTR(zline,VAL(SUBSTR(loczline,sel*2-1,2)),;
                     VAL(SUBSTR(lenzline,sel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'zmsg'+STR(sel,1) TO zmsgnum
 @ 5,1 SAY &zmsgnum

  I=0

* routine to deal with dbaseIII bug - not trapping leftarrow

  X=CHR(200)
  DO WHILE X=CHR(200)
   CALL INKEY WITH X
  ENDDO
  I=ASC(X)

  @ 16,5 SAY SPACE(44)

  DO CASE

   CASE I=28
    DO HENTPASS
```

```
CASE I=1
 sel=1

CASE I=6
 sel=7

CASE I=4  && right arrow
 IF sel=7
   sel=1
  ELSE
   sel=sel+1
 ENDIF

CASE I=19  && left arrow
 IF sel=1
   sel=7
  ELSE
   sel=sel-1
 ENDIF

CASE I=13  && return

 DO CASE

  CASE sel=1  && next
   IF .NOT. EOF()
    SKIP
   ELSE
    @ 16,5 SAY 'End of file - no more users!'
   ENDIF

 CASE sel=2  && previous
   IF .NOT. BOF()
    SKIP-1
   ELSE
    @ 16,5 SAY 'Start of file - no more users!'
   ENDIF

 CASE sel=3  && find
   GO TOP
   suser=SPACE(8)
   @ 11,15 SAY SPACE(8)
   @ 13,19 SAY ' '
   @ 9,5 SAY 'USER:' GET suser FUNCTION '!' PICTURE 'AAAAAAAA'
   READ
   IF READKEY()=36 .OR. READKEY()=292
    DO HENTPASS
    LOOP
   ENDIF
   SEEK suser
   IF .NOT. FOUND()
     @ 16,5 SAY 'User does not exist. Moved to start of file.'
     GO TOP
    ELSE
```

```
        @ 16,5 SAY 'User found - select  NEXT for others'
      ENDIF

    CASE sel=4 && add
      adding=.T.
      DO WHILE adding
       euser=SPACE(8)
       @ 11,15 SAY '        '
       @ 13,19 SAY ' '
       @ 9,11 GET euser FUNCTION '!' PICTURE 'AAAAAAAA'
       READ
       IF READKEY()=36 .OR. READKEY()=292
        DO HENTPASS
        LOOP
       ENDIF
       IF euser=SPACE(8)
        adding=.F.
        LOOP
       ENDIF
       APPEND BLANK
       REPLACE USER WITH euser
       @ 11,15 GET PASSWORD FUNCTION '!' PICTURE 'AAAAAAAA'
       READ
       IF READKEY()=36 .OR. READKEY()=292
        DO HENTPASS
        LOOP
       ENDIF
       notvalid=.T.  && to get value in range
       DO WHILE notvalid
        @ 13,19 GET ACCESS FUNCTION 'BZ' PICTURE '9'
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HENTPASS
         LOOP
        ENDIF
        IF ACCESS<1 .OR. ACCESS>4
         REPLACE  ACCESS  WITH  0  && only  way  to  blank  sinc
entering direct
         LOOP
        ENDIF
        notvalid=.F.
       ENDDO  && notvalid
      ENDDO  && adding

    CASE sel=5  && change
      @ 11,15 GET PASSWORD FUNCTION '!' PICTURE 'AAAAAAAA'
      READ
      IF READKEY()=36 .OR. READKEY()=292
       DO HENTPASS
       LOOP
      ENDIF
      notvalid=.T.  && to get value in range
      DO WHILE notvalid
       @ 13,19 GET ACCESS FUNCTION 'BZ' PICTURE '9'
       READ
```

```
      IF READKEY()=36 .OR. READKEY()=292
       DO HENTPASS
       LOOP
      ENDIF
      IF ACCESS<1 .OR. ACCESS>4
       REPLACE ACCESS WITH 0
       LOOP
      ENDIF
      notvalid=.F.
     ENDDO  && notvalid

   CASE sel=6  && delete
     DELETE
     del=.T.
     @ 16,5 SAY 'User deleted. Next user displayed.'
     SKIP
     IF EOF()
      @ 16,5 SAY 'Deleted. Moved to start of file.'
      GO TOP
     ENDIF

    CASE sel=7 && exit
      menuck=.F.
    ENDCASE

   OTHERWISE
    ? CHR(7)

  ENDCASE

 ENDDO

IF del
 PACK
ENDIF

USE

RETURN
```

```
*************************************************************
*                                                           *
*                        COLS.PRG                           *
*                                                           *
*                           by                              *
*                                                           *
*                     Geoff Minshull                        *
*                                                           *
*                                                           *
*©copyright Geoff Minshull, 1987                           *
*************************************************************

*
*  Called from: INIT
*

*
* This program deals with the installation of colours.  * Choices
are stored in COLS.MEM
*

CLEAR

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='change colours'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

cchoice=1  && controls which colour being amended, and loop

sel=1
I=0

STORE '011223324353' TO locxline  && location in xline
STORE '050503050404' TO lenxline  && length of items in xline

STORE  'Black      White      Red      Green      Blue       Exit
' TO xline
STORE 'Change text                  ' TO xmsg1
STORE 'Change background            ' TO xmsg2
STORE 'Change helpscreen text       ' TO xmsg3
STORE 'Change helpscreen background' TO xmsg4


DO WHILE cchoice<5

  SET COLOR TO &col1/&col2

  IF cchoice=2  .AND. I=13 && redraw backgound only if changed
   @ 0,0 CLEAR TO 11,79
  ENDIF
```

```
*
*  draw help text - leave on to illustrate effects of changes
*
 IF I=0 .OR.  (cchoice=3 .AND.  I=13) .OR. (cchoice=4 .AND. I=13)
  SET COLOR TO &col3/&col4
  @ 12,0 CLEAR TO 22,79 && clear box
  @ 12,1 TO 22,78 && draw border
  @  13,9  SAY  'You  can now change the  colours  used  in  this
program, by moving'
  @  14,9  SAY 'the highlight and  pressing RETURN on the  colour
required. The'
  @  15,9  SAY 'colours change immediately,  so you can  see  the
effect at once.'
  @ 16,9 SAY 'The colours are changed in this order: text'
  @ 17,48 SAY 'background'
  @ 18,48 SAY 'helpscreen text'
  @ 19,48 SAY 'helpscreen background'
  @  20,9  SAY  'You  CANNOT use the SAME  colour  for  text  AND
background!'
  @ 21,9 SAY 'When you are satisfied select EXIT'
 ENDIF

  IF I=13
   cchoice=cchoice+1
  ENDIF

  IF cchoice=5
   cchoice=1
  ENDIF

  SET COLOR TO &col1/&col2
  @ 1,40-(LEN(comname)*.5) SAY comname
  @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
  @ 4,1 SAY xline
  SET COLOR TO &col2/&col1
  col=VAL(SUBSTR(locxline,sel*2-1,2))
  hilite=SUBSTR(xline,VAL(SUBSTR(locxline,sel*2-1,2)),;
                      VAL(SUBSTR(lenxline,sel*2-1,2)))
  @ 4,col SAY hilite
  SET COLOR TO &col1/&col2

  STORE 'xmsg'+STR(cchoice,1) TO xmsgnum
  @ 6,1 SAY &xmsgnum
  SET COLOR TO &col2/&col1
  @ 24,0 SAY SPACE(80)
  @ 24,40-(LEN(bar)*.5) SAY bar
  @ 24,71 SAY DATE()
  SET              COLOR              TO              &col1/&col
  @ 23,0 SAY SPACE(80)

  I=0
  X=CHR(200)
  DO WHILE X=CHR(200)
   CALL INKEY WITH X
```

```
ENDDO
I=ASC(X)

DO CASE

CASE I=1
 sel=1

CASE I=6
 sel=6

CASE I=4  && right arrow
 IF sel=6
   sel=1
  ELSE
   sel=sel+1
 ENDIF

CASE I=19  && left arrow
 IF sel=1
   sel=6
  ELSE
   sel=sel-1
 ENDIF

CASE I=13
 DO CASE

  CASE sel=1

   DO CASE
    CASE cchoice=1
     IF col2<>'N'
      STORE 'N' TO col1  && text
     ENDIF

    CASE cchoice=2
     IF col1<>'N'
      STORE 'N' TO col2  && background
     ENDIF

    CASE cchoice=3
     IF col4<>'N'
      STORE 'N' TO col3  && helpscreen text
     ENDIF

    CASE cchoice=4
     IF col3<>'N'
      STORE 'N' TO col4  && helpscreen background
     ENDIF
   ENDCASE

  CASE sel=2
   DO CASE
```

```
   CASE cchoice=1
    IF col2<>'W'
     STORE 'W' TO col1
    ENDIF

   CASE cchoice=2
    IF col1<>'W'
     STORE 'W' TO col2
    ENDIF

   CASE cchoice=3
    IF col4<>'W'
     STORE 'W' TO col3
    ENDIF

   CASE cchoice=4
    IF col3<>'W'
     STORE 'W' TO col4
    ENDIF
  ENDCASE

 CASE sel=3
  DO CASE

   CASE cchoice=1
    IF col2<>'R'
     STORE 'R' TO col1
    ENDIF

   CASE cchoice=2
    IF col1<>'R'
     STORE 'R' TO col2
    ENDIF

   CASE cchoice=3
    IF col4<>'R'
     STORE 'R' TO col3
    ENDIF

   CASE cchoice=4
    IF col3<>'R'
     STORE 'R' TO col4
    ENDIF
  ENDCASE

 CASE sel=4
  DO CASE

   CASE cchoice=1
    IF col2<>'G'
     STORE 'G' TO col1
    ENDIF

   CASE cchoice=2
    IF col1<>'G'
```

```
          STORE 'G' TO col2
         ENDIF

        CASE cchoice=3
         IF col4<>'G'
          STORE 'G' TO col3
         ENDIF

        CASE cchoice=4
         IF col3<>'G'
          STORE 'G' TO col4
         ENDIF
       ENDCASE

     CASE sel=5
      DO CASE

       CASE cchoice=1
        IF col2<>'B'
         STORE 'B' TO col1
        ENDIF

       CASE cchoice=2
        IF col1<>'B'
         STORE 'B' TO col2
        ENDIF

       CASE cchoice=3
        IF col4<>'B'
         STORE 'B' TO col3
        ENDIF


       CASE cchoice=4
        IF col3<>'B'
         STORE 'B' TO col4
        ENDIF
       ENDCASE

    CASE sel=6    && exit option
      cchoice=5

    ENDCASE

  ENDCASE

ENDDO

* save colours

SAVE TO COLS ALL LIKE col?

SET COLOR TO &col1/&col2

RETURN
```

```
*************************************************************
*                                                           *
*                        ENTORD.PRG                         *
*                                                           *
*                             by                            *
*                                                           *
*                       Geoff Minshull                      *
*                                                           *
*                                                           *
*(c) copyright Geoff Minshull, 1987                         *
*************************************************************

*
*  Called from: ORDER
*

*
*  This program deals with order entry and maintenance.
* Note that orders cannot be deleted/changed after they
* have been processed.
*

CLEAR

SELECT PRODUCT
SET INDEX TO PDESC
SELECT CUSTOMER
SET ORDER TO 1
SELECT ORDER
SET ORDER TO 1

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))

l=13
menuck=.T.
sel=1
STORE '011124374960' TO locline
STORE '060506060404' TO lenline

STORE  'Singly     Batch          Delete         Change         List
Exit' TO line
STORE 'Enter orders one at a time          ' TO msg1
STORE 'Enter orders in batches             ' TO msg2
STORE 'Delete orders                       ' TO msg3
STORE 'Change order quantity               ' TO msg4
STORE 'List orders to screen or printer    ' TO msg5
STORE 'Return to previous menu             ' TO msg6

DO WHILE menuck
 SET COLOR TO &col1/&col2
 @ 4,3 SAY line
 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(locline,sel*2-1,2))+2
 hilite=SUBSTR(line,VAL(SUBSTR(locline,sel*2-1,2)),;
```

```
                    VAL(SUBSTR(lenline,sel*2-1,2)))
@ 4,col SAY hilite
SET COLOR TO &col1/&col2
STORE 'msg' + STR(sel,1) TO msgnum
@ 5,3 SAY &msgnum
@ 7,1 CLEAR TO 20,42    && clear screen

IF I=13
 bar='order entry menu'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,1 SAY '<F1> for help'
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
ENDIF

 I=0

 X=CHR(200)
 DO WHILE X=CHR(200)  && routine necessary because left arrow
  CALL INKEY WITH X   && not trapped by INKEY()
 ENDDO
 I=ASC(X)

 DO CASE

  CASE I=28  && help key
   IF sel=1 .OR. sel=2
    DO HENTORD
   ELSE
    DO HOAMEND
   ENDIF

  CASE I=1  && home key
   sel=1

  CASE I=6  && end key
   sel=6

  CASE I=4  && right arrow
   IF sel=6
     sel=1
    ELSE
     sel=sel+1
   ENDIF

  CASE I=19  && left arrow
   IF sel=1
     sel=6
    ELSE
     sel=sel-1
   ENDIF

  CASE I=13  && return
```

```
     DO CASE

      CASE sel=1  && singly
       bar='process one at a time'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,1 SAY '<F1> for help'
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
*
* first find customer
*

       looping=.T.   &&  main control loop for finding cust/prod
       @ 7,1 TO 20,42     && draw box

       DO WHILE looping
       cfind=.T.     &&  loop for finding customer
       pfind=.T.     &&  loop for finding product
       DO WHILE cfind
        SELECT CUSTOMER
        ename=SPACE(8)
        onum=0
        @  8,3 SAY 'Order num:    ' GET onum FUNCTION 'BZ'  PICTURE
'9999'
        @ 9,15 SAY SPACE(8)
        @ 10,15 SAY SPACE(20)
        @ 11,15 SAY SPACE(20)
        READ

        IF READKEY()=36 .OR. READKEY()=292
         DO HENTORD
         LOOP
        ENDIF

        IF onum=0
         cfind=.F.
         pfind=.F.
         looping=.F.
         LOOP
        ENDIF

        @ 9,3 SAY 'Short name: ' GET ename FUNCTION '!'
        @ 10,3  SAY 'Full name: '
        @ 11,3 SAY 'Address: '
        READ

        IF READKEY()=36 .OR. READKEY()=292
         DO HENTORD
         LOOP
        ENDIF

        ename=RTRIM(ename)
        SEEK ename
```

```
        IF .NOT. FOUND()
         @ 19,3 SAY 'Customer does not exist.                    '
         LOOP
        ENDIF

        next=.T.  && loop to confirm correct name
        DO WHILE next
         @ 9,15 SAY SHORTNAME
         @ 10,15  SAY FULLNAME
         @ 11,15 SAY ADD1
         entry=' '

         DO WHILE entry=' '
          @ 19,3  SAY 'Correct name? Y/N or X to quit: ' GET entry
FUNCTION '!'
          READ
          IF READKEY()=36 .OR. READKEY()=292
           DO HENTORD
           LOOP
          ENDIF
          @ 19,3 SAY SPACE(38)
          IF AT(entry,'NXY')=0
           entry=' '
           LOOP
          ENDIF
         ENDDO && entry

         IF entry='X'
          next=.F.
          LOOP
         ENDIF
         IF entry='Y'
          next=.F.
          cfind=.F.
         LOOP
         ENDIF
         SKIP
         IF EOF()
          @ 19,3 SAY 'No more names.  Press any key to quit '
          READ
          @ 19,3 SAY SPACE(38)
          next=.F.
          LOOP
         ENDIF
        ENDDO &&next
       ENDDO  && cfind
*
*  now find product
*
       DO WHILE pfind
        SELECT PRODUCT
        foundrec=.F.   &&  control for finding correct record
        edesc=SPACE(20)
        @ 13,3 SAY 'Description: ' GET edesc FUNCTION '!'
        READ
```

```
        IF READKEY()=36 .OR. READKEY()=292
         DO HENTORD
         LOOP
        ENDIF
        IF edesc=SPACE(20)
         pfind=.F.
        LOOP
        ENDIF
        edesc=RTRIM(edesc)
        SEEK edesc
        IF .NOT. FOUND()
         @ 19,3  SAY 'Product does not exist.                    '
         LOOP
        ENDIF
        next=.T.    && loop to confirm product

        DO WHILE next
         @ 13,17 SAY DESCRIP
         entry=' '

         DO WHILE entry=' '
          @ 19,3  SAY 'Correct product?  Y/N or X to quit:   '  GET
entry FUNCTION '!'
          READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HENTORD
          LOOP
         ENDIF
          @ 19,3 SAY SPACE(38)
          IF AT(entry,'NXY')=0
           entry=' '
           LOOP
          ENDIF
          ENDDO  && entry

         IF entry='X'
          next=.F.
          LOOP
         ENDIF
         IF entry='Y'
          next=.F.
          foundrec=.T.
          LOOP
         ENDIF
         SKIP
         IF EOF()
          @ 19,3 SAY 'No more products.Press any key to quit'
          READ
          @ 19,3 SAY SPACE(38)
          next=.F.
          LOOP
         ENDIF

        ENDDO && next
*
```

```
*  now enter quantity required
*
        IF foundrec
         quant=0   && order quantity
         @  15,3  SAY 'Enter quantity:  ' GET quant FUNCTION  'BZ'
PICTURE '999'
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HENTORD
          LOOP
         ENDIF
*
*  now append to order file
*
         SELECT ORDER
         SET ORDER TO 0
         APPEND BLANK
         REPLACE ONUMBER WITH onum
         REPLACE CUSTCODE WITH CUSTOMER->CUSTCODE
         REPLACE REFERENCE WITH PRODUCT->REFERENCE
         REPLACE OQUANT WITH quant
         REPLACE ODATE WITH DATE()  && system date
         @ 15,19 SAY '      '  && blank out order
       ENDIF

       ENDDO && pfind
       ENDDO && looping


      CASE sel=2  && batch

       bar='process in batches'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,1 SAY '<F1> for help'
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2

       batchentry=.T.

       DO WHILE batchentry
       batch=0       &&  items  in batch
       tot=0         &&  total quantities in batch
       totquant=0    && running total quantities
       ordcount=0    && running total orders entered
       looping=.T.   &&  main control loop for finding cust/prod
       @  8,3  SAY 'Enter number of items:   ' GET batch  FUNCTION
'BZ' PICTURE '99'
       @  9,3 SAY 'Enter total of quantities:  ' GET tot  FUNCTION
'BZ' PICTURE '99999'
       READ
       IF READKEY()=36 .OR. READKEY()=292
        DO HENTORD
        LOOP
```

```
        ENDIF
        @ 8,3 SAY SPACE(27)
        @ 9,3 SAY SPACE(32)
        @ 7,1 CLEAR TO 20,42  && clear anything on screen
        @ 7,1 TO 20,42    && draw box

        DO WHILE looping
        cfind=.T.     &&  loop for finding customer
        pfind=.T.     &&  loop for finding product

        DO WHILE cfind
         SELECT CUSTOMER
         ename=SPACE(8)
         onum=0
         @  8,3 SAY 'Order num:    ' GET onum FUNCTION 'BZ'  PICTURE
'9999'
         @ 9,15 SAY SPACE(8)
         @ 10,15 SAY SPACE(20)
         @ 11,15 SAY SPACE(20)
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HENTORD
          LOOP
         ENDIF
         IF onum=0
          cfind=.F.
          pfind=.F.
          looping=.F.
          LOOP
         ENDIF
         @ 9,3 SAY 'Short name: ' GET ename FUNCTION '!'
         @ 10,3  SAY 'Full name: '
         @ 11,3 SAY 'Address: '
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HENTORD
          LOOP
         ENDIF
         ename=RTRIM(ename)
         SEEK ename
         IF .NOT. FOUND()
          @ 19,3 SAY 'Customer does not exist.                  '
          LOOP
         ENDIF
         next=.T.  && loop to confirm correct name

         DO WHILE next
          @ 9,15 SAY SHORTNAME
          @ 10,15  SAY FULLNAME
          @ 11,15 SAY ADD1
          entry=' '

          DO WHILE entry=' '
           @ 19,3  SAY 'Correct name? Y/N or X to quit: ' GET entry
FUNCTION '!'
```

```
          READ
          IF READKEY()=36 .OR. READKEY()=292
           DO HENTORD
           LOOP
          ENDIF
          @ 19,3 SAY SPACE(38)
          IF AT(entry,'NXY')=0
           entry=' '
           LOOP
          ENDIF
         ENDDO && entry

         IF entry='X'
          next=.F.
          LOOP
         ENDIF
         IF entry='Y'
          next=.F.
          cfind=.F.
         LOOP
         ENDIF
         SKIP
         IF EOF()
          @ 19,3 SAY 'No more names.  Press any key to quit '
          READ
          @ 19,3 SAY SPACE(38)
          next=.F.
          LOOP
         ENDIF
        ENDDO &&next
       ENDDO  && cfind
*
*  now find product
*
       DO WHILE pfind
        SELECT PRODUCT
        foundrec=.F.   &&  control for finding correct record
        edesc=SPACE(20)
        @ 13,3 SAY 'Description: ' GET edesc FUNCTION '!'
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HENTORD
         LOOP
        ENDIF
        IF edesc=SPACE(20)
         pfind=.F.
        LOOP
        ENDIF
        edesc=RTRIM(edesc)
        SEEK edesc
        IF .NOT. FOUND()
         @ 19,3  SAY 'Product does not exist.                '
         LOOP
        ENDIF
        next=.T.    && loop to confirm product
```

```
        DO WHILE next
         @ 13,17 SAY DESCRIP
         entry=' '

         DO WHILE entry=' '
          @ 19,3  SAY 'Correct product?  Y/N or X to quit:  '  GET
entry FUNCTION '!'
          READ
          IF READKEY()=36 .OR. READKEY()=292
           DO HENTORD
           LOOP
          ENDIF
          @ 19,3 SAY SPACE(38)
          IF AT(entry,'NXY')=0
           entry=' '
           LOOP
          ENDIF
          ENDDO  && entry
         IF entry='X'
          next=.F.
          LOOP
         ENDIF
         IF entry='Y'
          next=.F.
          foundrec=.T.
          LOOP
         ENDIF
         SKIP
         IF EOF()
          @ 19,3 SAY 'No more products.Press any key to quit'
          READ
          @ 19,3 SAY SPACE(38)
          next=.F.
          LOOP
         ENDIF

        ENDDO && next
*
*   now enter quantity required
*
        IF foundrec
         quant=0   && order quantity
         @  15,3  SAY 'Enter quantity:  ' GET quant FUNCTION  'BZ'
PICTURE '999'
         READ
         IF READKEY()=36 .OR. READKEY()=292
          DO HENTORD
          LOOP
         ENDIF
         totquant=totquant+quant
         ordcount=ordcount+1
*
*   now append to order file
*
```

```
          SELECT ORDER
          SET ORDER TO 0
          APPEND BLANK
          REPLACE ONUMBER WITH onum
          REPLACE CUSTCODE WITH CUSTOMER->CUSTCODE
          REPLACE REFERENCE WITH PRODUCT->REFERENCE
          REPLACE OQUANT WITH quant
          REPLACE ODATE WITH DATE()  && system date
          @ 15,19 SAY '      '  && blank out order
        ENDIF

        ENDDO && pfind
        ENDDO && looping
        IF ordcount<>batch .OR.  tot<>totquant  && ie. batch totals
wrong
         @ 19,3 SAY 'Batch incorrect. Press key to re-enter'
         READ
         @ 19,3 SAY SPACE(38)
         SELECT ORDER
         SET ORDER TO 0
         GO BOTTOM
         SKIP-ordcount
         IF RECNO()<>1
          SKIP
         ENDIF
         DELETE NEXT ordcount
        ENDIF
         batchentry=.F.
      ENDDO && batchentry


      CASE sel=3  && delete records

       bar='delete orders'
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,1 SAY '<F1> for help'
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2
       SELECT ORDER
       SET ORDER TO 1
       REINDEX
       @ 7,1 TO 20,42
       search=.T.

       DO WHILE search
        @ 19,3 SAY SPACE(38)
        @ 8,15 SAY '     '
        @ 9,15 SAY SPACE(8)
        @ 10,15 SAY SPACE(20)
        @ 11,15 SAY SPACE(20)
        @ 13,15 SAY SPACE(8)
        @ 15,15 SAY '          '
        @ 19,3 SAY SPACE(38)
```

```
        onum=0
        @  8,3  SAY 'Order num:   ' GET onum FUNCTION 'BZ'  PICTURE
'9999'
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HOAMEND
         LOOP
        ENDIF
        IF onum=0
         search=.F.
         LOOP
        ENDIF
        SEEK onum
        IF .NOT. FOUND()
         @ 19,3 SAY 'Order not found. Press any key.'
         READ
         LOOP
        ENDIF
        IF DELETED()
         @ 19,3 SAY 'To be deleted. Press any key.   '
         READ
         LOOP
        ENDIF
         next=.T.

         DO WHILE next
          IF DELETED()
           SKIP
            IF EOF()
             @ 19,3 SAY 'No more orders. Press any key.    '
             READ
             @ 19,3 SAY SPACE (38)
             next=.F.
             ENDIF
           LOOP
          ENDIF
          reply=' '
          @ 8,15 SAY ONUMBER
          @ 9,3 SAY 'Short name: '
          @ 9,15 SAY CUSTOMER->SHORTNAME
          @ 10,3 SAY 'Full name: '
          @ 10,15 SAY CUSTOMER->FULLNAME
          @ 11,3 SAY 'Address: '
          @ 11,15 SAY CUSTOMER->ADD1
          @ 13,3 SAY 'Pr.code: '
          @ 13,15 SAY REFERENCE
          @ 15,3 SAY 'Quantity: '
          @ 15,15 SAY OQUANT
          @ 19,3 SAY 'N = next,  D = delete,  X = exit' GET  reply
FUNCTION '!'
          READ

          DO CASE

           CASE reply='N'
```

```
      SKIP
      IF EOF()
       @ 19,3 SAY 'No more orders. Press any key.    '
       READ
       @ 19,3 SAY SPACE (38)
       next=.F.
       ENDIF
      LOOP

     CASE reply='D'
      IF .NOT. UPDATED
       @ 19,3 SAY 'Order will be deleted on exit          '
       DELETE
      ELSE
       @ 19,3 SAY 'Record processed. Deletion not allowed'
      ENDIF
      READ
      next=.F.
      LOOP

     CASE reply='X'
      next=.F.
      LOOP
     OTHERWISE
      LOOP

    ENDCASE

   ENDDO  && next
 ENDDO  && search


CASE sel=4  && change

 bar='amend orders'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,1 SAY '<F1> for help'
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
 SELECT ORDER
 SET ORDER TO 1
 REINDEX
 @ 7,1 TO 20,42
 search=.T.

 DO WHILE search
  @ 19,3 SAY SPACE(38)
  @ 8,15 SAY '     '
  @ 9,15 SAY SPACE(8)
  @ 10,15 SAY SPACE(20)
  @ 11,15 SAY SPACE(20)
  @ 15,15 SAY '         '
  @ 19,3 SAY SPACE(38)
```

```
        onum=0
        @  8,3  SAY 'Order num:   ' GET onum FUNCTION 'BZ'  PICTURE
'9999'
        READ
        IF READKEY()=36 .OR. READKEY()=292
         DO HOAMEND
         LOOP
        ENDIF
        IF onum=0
         search=.F.
         LOOP
        ENDIF
        SEEK onum
        IF .NOT. FOUND()
         @ 19,3 SAY 'Order not found. Press any key.'
         READ
         LOOP
        ENDIF
        IF DELETED()
         @ 19,3 SAY 'To be deleted. Press any key.  '
         READ
         LOOP
        ENDIF
         next=.T.

         DO WHILE next
          IF DELETED()
           SKIP
            IF EOF()
             @ 19,3 SAY 'No more orders. Press any key.    '
             READ
             @ 19,3 SAY SPACE (38)
             next=.F.
             ENDIF
           LOOP
          ENDIF
          reply=' '
          @ 8,15 SAY ONUMBER
          @ 10,3 SAY 'Pr.code: '
          @ 10,15 SAY REFERENCE
          @ 15,3 SAY 'Quantity: '
          @ 15,15 SAY OQUANT FUNCTION 'BZ'
          @ 19,3 SAY 'N = next,  C = change,  X = exit' GET  reply
FUNCTION '!'
          READ

          DO CASE

           CASE reply='N'
            SKIP
            IF EOF()
             @ 19,3 SAY 'No more orders. Press any key   '
             READ
             @ 19,3 SAY SPACE (38)
             next=.F.
```

```
        ENDIF
       LOOP

      CASE reply='C'
       IF .NOT. UPDATED
        @ 15,15 SAY '    '
        @ 15,15 GET OQUANT FUNCTION 'BZ' PICTURE '999'
        READ
       ELSE
        @ 19,3 SAY 'Already processed. Change not allowed'
        READ
        @ 19,3 SAY SPACE(38)
       ENDIF
       next=.F.
       LOOP

      CASE reply='X'
       next=.F.
       LOOP
      OTHERWISE
       LOOP

     ENDCASE

    ENDDO  && next
  ENDDO  && search


CASE sel=5  && list

 bar='list orders'
 SET COLOR TO &col2/&col1
 @ 24,0 SAY SPACE(80)
 @ 24,40-(LEN(bar)*.5) SAY bar
 @ 24,71 SAY DATE()
 SET COLOR TO &col1/&col2
 SELECT ORDER
 SET ORDER TO 1
 REINDEX
 @ 7,1 TO 20,42
 GO TOP
 reply='N'
 @ 19, 3 SAY 'Print orders? Y/N ' GET reply FUNCTION '!'
 READ

 IF reply='Y'
   @ 19,3 SAY 'Press any key when printer ready'
   READ
   @ 19,3 SAY 'Printing.....                    '
   SET DEVICE TO PRINT
   EJECT
   @ 2,40-(LEN(comname)*.5) SAY comname
   @ 2,65 SAY DATE()
   @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
   @ 5,32 SAY 'List of orders'
```

```
    @ 6,32 SAY '--------------'
    @ 8,3 SAY 'Order'
    @ 8,10 SAY 'Short name'
    @ 8,27 SAY 'Address'
    @ 8,47 SAY 'Pr.code'
    @ 8,72 SAY 'Quantity'
    linecount=10  && initialise printer line count
    DO WHILE .NOT. EOF()
     IF DELETED()
      SKIP
      LOOP
     ENDIF
     linecount=linecount+1
     IF linecount=50
      linecount=10
      EJECT
      @ linecount,3 SAY 'Order'
      @ linecount,10 SAY 'Short name'
      @ linecount,27 SAY 'Address'
      @ linecount,47 SAY 'Pr.code'
      @ linecount,72 SAY 'Quantity'
      linecount=linecount+2
     ENDIF
     @ linecount,3 SAY ONUMBER
     key=CUSTCODE
     SELECT CUSTOMER
     SET ORDER TO 2
     SEEK (key)
     SET ORDER TO 1
     SELECT ORDER
     @ linecount,10 SAY CUSTOMER->SHORTNAME
     @ linecount,22 SAY CUSTOMER->ADD1
     @ linecount,47 SAY REFERENCE
     @ linecount,72 SAY OQUANT
     SKIP
    ENDDO  && eof()
    EJECT
    SET DEVICE TO SCREEN
    @ 19,3 SAY 'Printing complete.'

ELSE

     @ 8,3 SAY 'Order num: '
     @ 9,3 SAY 'Short name: '
     @ 10,3 SAY 'Address: '
     @ 12,3 SAY 'Pr.code: '
     @ 15,3 SAY 'Quantity: '

   DO WHILE .NOT. EOF()
     IF DELETED()
      SKIP
      LOOP
     ENDIF
     @ 8,15 SAY ONUMBER
     key=CUSTCODE
```

```
            SELECT CUSTOMER
            SET ORDER TO 2
            SEEK (key)
            SET ORDER TO 1
            SELECT ORDER
            @ 9,15 SAY CUSTOMER->SHORTNAME
            @ 10,15 SAY CUSTOMER->ADD1
            @ 12,15 SAY REFERENCE
            @ 15,15 SAY OQUANT
            @ 19,3 SAY 'Press any key to continue'
            READ
            SKIP

          ENDDO  && EOF()

        ENDIF  && reply


       CASE sel=6  && exit
        menuck=.F.

      ENDCASE

     OTHERWISE

      ? CHR(7)

    ENDCASE

  ENDDO

SELECT ORDER
PACK   && incorrectly batched entries deleted here
CLEAR
menuck=.T.  && so doesn't fall thro calling menu
sel=1  && choice in calling menu.
SELECT PRODUCT
SET INDEX TO

RETURN
```

```
*****************************************************************
*                                                               *
*                        STAT.PRG                               *
*                                                               *
*                             by                                *
*                                                               *
*                      Geoff Minshull                           *
*                                                               *
*                                                               *
*(c)copyright Geoff Minshull, 1987                              *.
*****************************************************************

*
*  Called from: REPORTS
*

*
*  This program prints statements using the order, customer and
* payment files.
*


      choice=' '

      DO WHILE choice<>'X'
      SELECT CUSTOMER
      SET ORDER TO 2
      SELECT 10
      USE PAYMENT INDEX PCODE
      GO TOP
      SELECT ORDER
      SET ORDER TO 2
      GO TOP
      choice=' '
      noorder=.t.  && test if any statements

      @ 0,0 CLEAR TO 23,79
      @ 3,1 SAY 'You can get a Screen report  -  S'
      @ 4,1 SAY '         or a  Printed report -  P'
      @ 5,1 SAY '         or eXit              -  X'

      DO WHILE choice<>'S' .AND. choice<>'P' .AND. choice<>'X'
       @ 3,40 SAY 'Enter your choice: ' GET choice FUNCTION '!'
       READ
       IF READKEY()=36
        DO HSTAT
       ENDIF
      ENDDO

      DO CASE

       CASE choice='S'
        tot=0
        tinvtot=0
        linecount=10
```

```
tcust=0  && test for change in customer
firstpass=.T.

DO WHILE !.NOT. EOF()
 IF INVOICED
  IF firstpass
   tinvnum=INVNUM
  ENDIF
  IF tcust<>CUSTCODE
   IF .NOT. firstpass
    SELECT 10
    SEEK key
    tpcode=CUSTCODE

    DO WHILE tpcode=CUSTCODE .AND. (.NOT. EOF())
     linecount=linecount+1
     IF linecount=21
      @ 23,1 SAY 'Press any key....'
      READ
      linecount=9
      @ 11,0 CLEAR TO 23,79
     ENDIF && linecount
     @ linecount,34 SAY tinvtot
     linecount=linecount+1
     @ linecount,0 SAY PDATE
     @ linecount,20 SAY INVNUM
     @ linecount,60 SAY AMOUNT
     tot=tot-AMOUNT
     SKIP
    ENDDO  && tpcode=custcode

    IF  tpcode=CUSTCODE  .AND.  EOF()  && print total  o
last item
     @ linecount+1,34 SAY tinvtot
     linecount=linecount+1
    ENDIF

    tinvtot=0
    tinvnum=INVNUM
    @ linecount+2,25 SAY 'Balance: '
    @ linecount+2,34 SAY tot
    tot=0
    linecount=10
    READ
    SELECT ORDER
   ENDIF  &&  not firstpass
   firstpass=.F.
   key=CUSTCODE
   noorder=.f.  && ie. there are some valid statements
   SELECT CUSTOMER
   SEEK key
   @ 8,0 CLEAR TO 23,79
   @ 8,0 SAY 'Statement for: '
   @ 8,15 SAY FULLNAME
   @ 8,40 SAY TOWN
```

```
            SELECT ORDER
            @    10,0    SAY    'Inv./pay    date            Inv.     no
Due                  Paid'
          ENDIF && tcust<>custcode
          linecount=linecount+1
          IF linecount=21
           @ 23,1 SAY 'Press any key....'
           READ
           linecount=9
           @ 11,0 CLEAR TO 23,79
          ENDIF && linecount
          IF INVNUM<>tinvnum
           IF tinvtot<>0
            @ linecount,34 SAY tinvtot
           ENDIF
           linecount=linecount+1
           tinvtot=0
           tinvnum=INVNUM
          ENDIF
           @ linecount,0 SAY INVDATE
           @ linecount,20 SAY INVNUM
           @ linecount,40 SAY ORDTOT
          tinvtot=tinvtot+ORDTOT
          tot=tot+ORDTOT
          tcust=CUSTCODE
         ENDIF   &&  invoiced
         SKIP
         ENDDO  eof
*
*  jump out if no statements
*
         IF noorder
          @  10,0  SAY 'You must issue invoices  before  producin
statements!'
          READ
          @                  10,0                SAY
   ,
          LOOP
         ENDIF
*
*  process statements
*
         SELECT 10
         SEEK key
         tpcode=CUSTCODE

         DO WHILE tpcode=CUSTCODE .AND. (.NOT. EOF())
          linecount=linecount+1
          IF linecount=21
           @ 23,1 SAY 'Press any key....'
           READ
           linecount=9
           @ 11,0 CLEAR TO 23,79
          ENDIF && linecount
          @ linecount,34 SAY tinvtot
```

```
          linecount=linecount+1
          @ linecount,0 SAY PDATE
          @ linecount,20 SAY INVNUM
          @ linecount,60 SAY AMOUNT
          tot=tot-AMOUNT
          SKIP
         ENDDO  && tpcode=custcode

         tinvtot=0
         tinvnum=INVNUM
         @ linecount+2,25 SAY 'Balance: '
         @ linecount+2,34 SAY tot
         READ

        CASE choice='P'
         tot=0
         tinvtot=0
         linecount=16
         @ 19,3 SAY 'Press any key when printer ready.'
         READ
         @ 9,0 CLEAR TO 23,79
         @ 19,3 SAY 'Printing.....'
         SET DEVICE TO PRINT
         EJECT
         tcust=0  && test for change in customer
         firstpass=.T.

         DO WHILE .NOT. EOF()
          IF INVOICED
           IF firstpass
            tinvnum=INVNUM
           ENDIF
           IF tcust<>CUSTCODE
            IF .NOT. firstpass
             SELECT 10
             SEEK key
             tpcode=CUSTCODE

             DO WHILE tpcode=CUSTCODE .AND. (.NOT. EOF())
              linecount=linecount+1
              IF linecount=50
               linecount=5
               EJECT
               @   3,0   SAY   'Inv./pay   date          Inv.    n
Due                 Paid'
              ENDIF && linecount
              @ linecount,34 SAY tinvtot
              linecount=linecount+1
              @ linecount,0 SAY PDATE
              @ linecount,20 SAY INVNUM
              @ linecount,60 SAY AMOUNT
              tot=tot-AMOUNT
              SKIP
             ENDDO  && tpcode=custcode
```

```
            IF  tpcode=CUSTCODE  .AND.  EOF()  && print total  o
last item
             @ linecount+1,34 SAY tinvtot
             linecount=linecount+1
            ENDIF
            tinvtot=0
            tinvnum=INVNUM
            @ linecount+2,25 SAY 'Balance: '
            @ linecount+2,34 SAY tot
            tot=0
            linecount=16
            EJECT
            SELECT ORDER
           ENDIF  &&  not firstpass
           firstpass=.F.
           key=CUSTCODE
           noorder=.f.  && ie. there are some valid statements
           SELECT CUSTOMER
           SEEK key
           @ 4,35 SAY 'STATEMENT'
           @ 4,60 SAY DATE()
           @ 7,8 SAY comname
           @ 7,50 SAY 'TO: '
           @ 7,55 SAY FULLNAME
           @ 8,8 SAY comadd1
           @ 8,55 SAY ADD1
           @ 9,8 SAY comtown
           @ 9,55 SAY ADD2
           @ 10,8 SAY compcode
           @ 10,55 SAY ADD3
           @ 11,8 SAY 'Phone: '
           @ 11,15 SAY comphone
           @ 11,55 SAY TOWN
           @ 12,8 SAY 'VAT No: '
           @ 12,16 SAY comvat
           SELECT ORDER
           @   15,0   SAY   'Inv./pay   date          Inv.     no
Due              Paid'
          ENDIF && tcust<>custcode
          linecount=linecount+1
          IF linecount=50
           linecount=5
           EJECT
           @    3,0    SAY   'Inv./pay   date          Inv.     no
Due              Paid'
          ENDIF && linecount
          IF INVNUM<>tinvnum
           IF tinvtot<>0
            @ linecount,34 SAY tinvtot
           ENDIF
           linecount=linecount+1
           tinvtot=0
           tinvnum=INVNUM
          ENDIF
          @ linecount,0 SAY INVDATE
```

```
          @ linecount,20 SAY INVNUM
          @ linecount,40 SAY ORDTOT
          tinvtot=tinvtot+ORDTOT
          tot=tot+ORDTOT
          tcust=CUSTCODE
         ENDIF   &&  invoiced
         SKIP
         ENDDO  eof
*
*  jump out if no statements
*
         IF noorder
          SET DEVICE TO SCREEN
          @ 19,3 SAY SPACE(35)
          @  10,0  SAY 'You must issue invoices  before  producing
statements!'
          READ
          @                  10,0                SAY                 '
   '
          LOOP
         ENDIF
*
*  process statements
*
         SELECT 10
         SEEK key
         tpcode=CUSTCODE

         DO WHILE tpcode=CUSTCODE .AND. (.NOT. EOF())
          linecount=linecount+1
          IF linecount=50
           linecount=5
           EJECT
           @    3,0   SAY   'Inv./pay    date          Inv.     no:
Due              Paid'
          ENDIF && linecount
          @ linecount,34 SAY tinvtot
          linecount=linecount+1
          @ linecount,0 SAY PDATE
          @ linecount,20 SAY INVNUM
          @ linecount,60 SAY AMOUNT
          tot=tot-AMOUNT
          SKIP
         ENDDO  && tpcode=custcode

         tinvtot=0
         tinvnum=INVNUM
         @ linecount+2,25 SAY 'Balance: '
         @ linecount+2,34 SAY tot
         EJECT
         SET DEVICE TO SCREEN

        CASE choice='X'
         SELECT 10
         USE
```

```
 LOOP
ENDCASE
ENDDO    && choice<>'X'

RETURN
```

```
*****************************************************************
*                                                               *
*                        REOMAX.PRG                             *
*                                                               *
*                              by                               *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*(c)copyright Geoff Minshull, 1987                              *
*****************************************************************

*
*  Called from: REPORTS
*

*
*  This program is for printing stock lists under minimum
*  and over maximum stock
*

SELECT PRODUCT
SET INDEX TO PREF
choice=' '

DO WHILE choice<>'X'
 GO TOP
 none=.T.  && test for no items
 choice=' '
 @ 0,0 CLEAR TO 23,79
 @ 3,1 SAY 'You can get a Screen report  -  S'
 @ 4,1 SAY '         or a Printed report  -  P'
 @ 5,1 SAY '         or eXit              -  X'

 DO WHILE choice<>'S' .AND. choice<>'P' .AND. choice<>'X'
  @ 3,40 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
  IF READKEY()=36
   DO HREOMAX
  ENDIF
 ENDDO

 DO CASE

  CASE choice='S'
   linecount=10
   @ 10,0 CLEAR TO 23,79
   @  8,0  SAY  'Code  Description            QIS       Min   Max
Re-order  Supplier  Difference'
   @  9,0 SAY '                                        stock  stock
quant.     code'

   DO WHILE .NOT. EOF()
    IF max
     IF QIS<=MAXSTOCK
```

```
        SKIP
        LOOP
       ENDIF
       ELSE
        IF QIS>=MINSTOCK
         SKIP
         LOOP
        ENDIF
      ENDIF && max
      linecount=linecount+1
      IF linecount=22
       @ 23,1 SAY 'Press any key....'
       READ
       linecount=11
       @ 10,0 CLEAR TO 23,79
      ENDIF  && linecount
      none=.F.
      @ linecount,0 SAY REFERENCE
      @ linecount,6 SAY DESCRIP
      @ linecount,29 SAY QIS
      @ linecount,38 SAY MINSTOCK
      @ linecount,43 SAY MAXSTOCK
      @ linecount,49 SAY ROQ
      @ linecount,59 SAY SUPPCODE
      IF max
       @ linecount,69 SAY  QIS-MAXSTOCK
      ELSE
       @ linecount,69 SAY MINSTOCK-QIS
      ENDIF
      SKIP
      ENDDO  && EOF

      IF none
       @ linecount+1,22 SAY 'There are no items in this category'
      ENDIF
      READ

    CASE choice='P'
     linecount=10
     @ 19,3 SAY 'Press any key when printer ready.'
     READ
     @ 10,0 CLEAR TO 23,79
     @ 19,3 SAY 'Printing.....'
     SET DEVICE TO PRINT
     EJECT
     @ 2,40-(LEN(comname)*.5) SAY comname
     @ 2,65 SAY DATE()
     @ 3,40-(LEN(comname)*.5) SAY REPLICATE('-',LEN(comname))
     @  8,0  SAY  'Code  Description            QIS       Min   Max
Re-order  Supplier  Difference'
     @  9,0 SAY '                                        stock  stock
quant.     code'

     DO WHILE .NOT. EOF()
      IF max
```

```
        IF QIS<=MAXSTOCK
         SKIP
         LOOP
        ENDIF
        ELSE
         IF QIS>=MINSTOCK
          SKIP
          LOOP
         ENDIF
       ENDIF && max
       linecount=linecount+1
       IF linecount=50
        linecount=10
        EJECT
       ENDIF  && linecount
       none=.F.
       @ linecount,0 SAY REFERENCE
       @ linecount,6 SAY DESCRIP
       @ linecount,29 SAY QIS
       @ linecount,38 SAY MINSTOCK
       @ linecount,43 SAY MAXSTOCK
       @ linecount,49 SAY ROQ
       @ linecount,59 SAY SUPPCODE
       IF max
        @ linecount,69 SAY  QIS-MAXSTOCK
       ELSE
        @ linecount,69 SAY MINSTOCK-QIS
       ENDIF
       SKIP
       ENDDO  && EOF

       IF none
        @ linecount+1,22 SAY 'There are no items in this category'
       ENDIF
       EJECT
       SET DEVICE TO SCREEN

   CASE choice='X'
    LOOP
   ENDCASE
   ENDDO  && choice<>'X'

RETURN
```

```
*****************************************************************
*                                                               *
*                        BARCHART.PRG                           *
*                                                               *
*                             by                                *
*                                                               *
*                       Geoff Minshull                          *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called from: REPORTS
*

*
*  This program produces 2 bar charts - product valuation at
* costprice; and sales at saleprice. Both charts use product
* groups.
*

CLEAR
bar='bar graphs'

SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(79)
@ 24,40-(LEN(bar)*.5) SAY bar
SET COLOR TO &col1/&col2

valuation=.t.

choice=' '
DO WHILE choice<>'X'
 @ 0,0 CLEAR TO 23,79
 @ 1,40-(LEN(comname)*.5) SAY comname
 @ 2,40-(LEN(comname)*.5) SAY REPLICATE('-',LEN(comname))
 choice=' '
 @ 5,1 SAY 'Stock valuation by product group at Cost price - C'
 @ 6,1 SAY '       or sales by product group at Sale price - S'
 @ 7,1 SAY '       or eXit                                 - X'

 DO WHILE choice<>'C' .AND. choice<>'S' .AND. choice<>'X'
  @ 5,53 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
 ENDDO

 IF choice='C'
  valuation=.T.
 ENDIF
 IF choice='S'
  valuation=.F.
 ENDIF
 IF choice='X'
  LOOP
```

```
 ENDIF

 @ 0,0 CLEAR TO 23,79

 IF valuation
  SELECT PRODUCT
  SET INDEX TO PGROUP
 ELSE
  SELECT 10
  USE PRODHIST INDEX HGROUP
 ENDIF

 GO TOP
 STORE  0  TO  g0,g1,g2,g3,g4,g5,g6,g7,g8,g9  &&  initialise  ba
heights
 tgroup=GROUP
 gcost=0
 maxtot=0  && maximum total

 DO WHILE .NOT. EOF()   && store values to variables g0 to g9
  IF GROUP<>tgroup
   var='g'+LTRIM(STR(tgroup))  && gives variable name
   STORE gcost TO &var
   IF gcost>maxtot
    maxtot=gcost
   ENDIF
   gcost=0
  ENDIF

  IF valuation
   gcost=gcost+QIS*COSTPRICE
  ELSE
   gcost=gcost+SALES*SALEPRICE
  ENDIF
  tgroup=GROUP
  SKIP
 ENDDO  && eof

 IF gcost>maxtot
  maxtot=gcost
 ENDIF

 var='g'+LTRIM(STR(tgroup))  && gives variable name
 STORE gcost TO &var

* draw xaxis

 @ 22,5 SAY '+'
 @ 22,6 SAY REPLICATE('-----+',10)
 @ 23,0 SAY 'Group:'
 @  23,9  SAY '0    1     2     3     4     5     6     7
9'

 steps=maxtot/20   && get scale - 20=num available lines
 steps=INT(steps+.5)
```

```
* draw yaxis

 @ 22,4 SAY '0'
 looper=21
 legend=steps*5
 DO WHILE looper >= 2
  IF looper=2 .OR. looper=7 .OR. looper=12 .OR. looper=17
   @ looper,0 SAY LTRIM(STR(legend))
   @ looper,5 SAY '+'
   legend=legend+steps*5
  ELSE
   @ looper,5 SAY CHR(179)
  ENDIF
  looper=looper-1
 ENDDO
 @ 0,0 SAY 'Value'

* draw bars

 looper=0
 col=8
 DO WHILE looper<10
  var='g'+LTRIM(STR(looper))
  IF &var=0
   looper=looper+1
   col=col+6
   LOOP
  ENDIF
  height=INT((&var/steps)+.5)
  row=21
  DO WHILE height>0
   @ row,col SAY CHR(219)
   row=row-1
   height=height-1
  ENDDO
  @ row,col SAY LTRIM(STR(&var))
  looper=looper+1
  col=col+6
 ENDDO && looper<10

 READ  && pause while key pressed

 ENDDO choice<>'x'

SELECT PRODUCT
SET INDEX TO
IF .NOT. valuation
  SELECT 10
  USE
ENDIF
RETURN
```

```
*****************************************************************
*                                                               *
*                        LEDGER.PRG                             *
*                                                               *
*                             by                                *
*                                                               *
*                      Geoff Minshull                           *
*                                                               *
*                                                               *
*©copyright Geoff Minshull, 1987                                *
*****************************************************************

*
*  Called from: REPORTS
*

*
*  This program produces the sales ledger.
*


CLEAR

bar='sales ledger'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

SELECT CUSTOMER
SET ORDER TO 2
SELECT ORDER
SET INDEX TO ODATE
REINDEX
GO TOP

choice=' '
DO WHILE choice<>'X'
 choice=' '
 @ 3,1 SAY 'You can get a Screen report - S'
 @ 4,1 SAY '         or a Printed report - P'
 @ 5,1 SAY '         or eXit             - X'

 DO WHILE choice<>'S' .AND. choice<>'P' .AND. choice<>'X'
  @ 3,40 SAY 'Enter your choice: ' GET choice FUNCTION '!'
  READ
  IF READKEY()=36
   @ 8,0 CLEAR TO 23,79
   DO HLEDGER  && help screen
  ENDIF
 ENDDO

 DO CASE
```

```
  CASE choice='S'
   GO TOP
   @ 8,0 CLEAR TO 23,79
   linecount=8
   tdate=SPACE(10)  && test for month change
   tot=0  && month total
   gtot=0 && grand total
   @  8,0  SAY  'Date         Name                                    Inv.no:
total ex.VAT'

   DO WHILE .NOT. EOF()
    IF INVOICED
     linecount=linecount+1
     IF linecount=22
      @ 23,1 SAY 'Press any key....'
      READ
      linecount=9
      @ 9,0 CLEAR TO 23,79
     ENDIF  && linecount
     IF tdate<>CMONTH(INVDATE)
      IF tot<>0  && so first time not printed
       @ linecount,33 SAY 'Total for '
       @ linecount,43 SAY tdate
       @ linecount,54 SAY tot
       tot=0
       linecount=linecount+2
      ENDIF  && tot
     ENDIF && tdate
     key=CUSTCODE
     SELECT CUSTOMER
     SEEK key
     SELECT ORDER
     @ linecount,0 SAY INVDATE
     @ linecount,11 SAY CUSTOMER->FULLNAME
     @ linecount,39 SAY INVNUM
     @ linecount,54 SAY ORDTOT*100/115
     tot=tot+(ORDTOT*100/115)
     gtot=gtot+(ORDTOT*100/115)
     tdate=CMONTH(INVDATE)
    ENDIF  && invoiced
    SKIP
   ENDDO  && eof

   IF linecount>20
    linecount=10
    @ 9,0 CLEAR TO 23,79
   ENDIF
   @ linecount+1,33 SAY 'Total for '
   @ linecount+1,43 SAY tdate
   @ linecount+1,54 SAY tot
   @ linecount+3,33 SAY 'Grand total: '
   @ linecount+3,54 SAY gtot
   READ
```

```
   CASE choice='P'
    @ 8,0 CLEAR TO 23,79
    @ 19,3 SAY 'Press any key when printer ready.'
    READ
    @ 19,3 SAY SPACE(34)
    @ 19,3 SAY 'Printing.....'
    SET DEVICE TO PRINT
    EJECT
    tot=0
    gtot=0
    linecount=8
    tdate=SPACE(10)
    @ 2,40-(LEN(comname)*.5) SAY comname
    @ 2,65 say DATE()
    @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
    @ 5,32 SAY 'Sales ledger'
    @ 6,32 SAY '------------'
    @  8,0  SAY 'Date        Name                                    Inv.no:
total ex.VAT'
    GO TOP

    DO WHILE .NOT. EOF()
     IF INVOICED
      linecount=linecount+1
      IF linecount=50
       EJECT
       linecount=10
      ENDIF  && linecount
      IF tdate<>CMONTH(INVDATE)
       IF tot<>0  && so first time not printed
        @ linecount,33 SAY 'Total for '
        @ linecount,43 SAY tdate
        @ linecount,54 SAY tot
        tot=0
        linecount=linecount+2
       ENDIF  && tot
      ENDIF && tdate
      key=CUSTCODE
      SELECT CUSTOMER
      SEEK key
      SELECT ORDER
      @ linecount,0 SAY INVDATE
      @ linecount,11 SAY CUSTOMER->FULLNAME
      @ linecount,39 SAY INVNUM
      @ linecount,54 SAY ORDTOT*100/115
      tot=tot+(ORDTOT*100/115)
      gtot=gtot+(ORDTOT*100/115)
      tdate=CMONTH(INVDATE)
      REPLACE LEDGERED WITH .T.
     ENDIF  && invoiced
     SKIP
    ENDDO  && eof

    @ linecount+1,33 SAY 'Total for '
    @ linecount+1,43 SAY tdate
```

```
      @ linecount+1,54 SAY tot
      @ linecount+3,33 SAY 'Grand total: '
      @ linecount+3,54 SAY gtot
      EJECT
      SET DEVICE TO SCREEN
      @ 19,3 SAY 'Print complete'

    CASE choice='X'
     LOOP
    ENDCASE

ENDDO  && choice='x'

SELECT ORDER
SET INDEX TO ONUM,OCUST

RETURN
```

```
*************************************************************
*                                                           *
*                        AUDIT.PRG                          *
*                                                           *
*                             by                            *
*                                                           *
*                       Geoff Minshull                      *
*                                                           *
*                                                           *
*©copyright Geoff Minshull, 1987                           *
*************************************************************

*
*  Called from: REPORTS
*

*
*  This produces the audit trails - one for payments, one for
* orders. Therefore uses ORDER and PAYMENT files. It is not
* possible - in this limited system - to devise a way of
* automatically deleting old orders/payments. Therefore, after
* trailing,  user is given option of deleting the records. Note
* after  deletion,  ENTRY  WILL  NOT APPEAR  ON  STATEMENTS.  In
* practice,therefore, it is likely that only the lecturer will do
* deletions when they are starting a fresh sequence
* of operations with a class.
* The same records, even if not deleted, will not appear on > 1
* trail.
* In the order file, unless UPDATED, INVOICED, TRAILED all are
* true, a record will NOT be deleted.
* A record will not appear on the order trail unless UPDATED
* and INVOICED are both true. (UPDATED refers to master files)
*

      CLEAR
      @ 1,40-(LEN(comname)*.5) SAY comname
      @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
      bar='print audit trail'
      SET COLOR TO &col2/&col1
      @ 24,0 SAY SPACE(80)
      @ 24,1 SAY '<F1> for help'
      @ 24,40-(LEN(bar)*.5) SAY bar
      @ 24,71 SAY DATE()
      SET COLOR TO &col1/&col2

      SELECT CUSTOMER
      SET ORDER TO 2
      SELECT ORDER
      SET ORDER TO 1  && order number index
      GO TOP

      linecount=9
      choice=' '

      DO WHILE choice<>'X'
```

```
        SELECT ORDER
        GO TOP
        choice=' '
        @ 5,1 SAY 'You can Print the audit trail - P'
        @ 6,1 SAY '                          or eXit - X'
        DO WHILE choice<>'P' .AND. choice<>'X'
         @ 5,42 SAY 'Enter your choice: ' GET choice FUNCTION '!'
         READ
         @ 10,1 CLEAR TO 11,50
         IF READKEY()=36
          DO HAUDIT
         ENDIF
        ENDDO

        DO CASE

         CASE choice='P'
          @ 10,1 SAY 'Printing.....              '
          SET DEVICE TO PRINT
          EJECT
          @ 2,40-(LEN(comname)*.5) SAY comname
          @ 2,65 SAY DATE()
          @     3,40-(LEN(comname)*.5)    SAY    REPLICATE     ('-
',LEN(comname))
          @ 5,26 SAY 'Audit trail - orders'
          @ 6,26 SAY '--------------------'
          @  8,0 SAY 'Fullname                Prod.  code   Quantity
Order no:     Inv. no:'

          DO WHILE .NOT. EOF()
           IF linecount=50
            EJECT
            @ 8,0 SAY 'Fullname                Prod.  code  Quantity
Order no:     Inv. no:'
            linecount=9
           ENDIF
           IF (UPDATED .AND. INVOICED .AND. (.NOT. TRAILED))
            key=CUSTCODE
            SELECT CUSTOMER
            SEEK key
            SELECT ORDER
            @ linecount,0 SAY CUSTOMER->FULLNAME
            @ linecount,22 SAY REFERENCE
            @ linecount,34 SAY OQUANT
            @ linecount,45 SAY ONUMBER
            @ linecount,58 SAY INVNUM
            linecount=linecount+1
            REPLACE TRAILED WITH .T.
           ENDIF  && updated and invoiced
           SKIP
          ENDDO  && eof
          EJECT
*
*   now do payments file
*
```

```
        SELECT 10
        USE PAYMENT INDEX PCODE
        linecount=9
        @ 2,40-(LEN(comname)*.5) SAY comname
        @ 2,65 SAY DATE()
        @ 3,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
        @ 5,26 SAY 'Audit trail - payments'
        @ 6,26 SAY '----------------------'
        @ 8,0 SAY 'Fullname                Inv. no:          Payment
date          Amount'

        DO WHILE .NOT. EOF()
         IF linecount=50
          EJECT
          @    8,0     SAY   'Fullname                    Inv.     no:
Payment date           Amount'
          linecount=9
         ENDIF
         IF .NOT. TRAILED
          key=CUSTCODE
          SELECT CUSTOMER
          SEEK key
          SELECT 10
          @ linecount,0 SAY CUSTOMER->FULLNAME
          @ linecount,23 SAY INVNUM
          @ linecount,40 SAY PDATE
          @ linecount,59 SAY AMOUNT
          linecount=linecount+1
          REPLACE TRAILED WITH .T.
         ENDIF  && not trailed
         SKIP
        ENDDO  && eof
        EJECT
        SET DEVICE TO SCREEN
*
*  now give option to erase records
*
        echoice='N'
        @ 23,5 SAY '                                          '
        @ 10,1 SAY 'Printing now complete. '
        @ 11,1 SAY 'Delete old records from the order file? Y/N '
GET echoice FUNCTION '!'
        READ
        IF echoice='Y'
         @ 11,1 SAY SPACE(50)
         @ 11,1 SAY 'Deleting now......'
         SELECT ORDER
         GO TOP
         DO WHILE .NOT. EOF()
          IF  (INVOICED  .AND.   UPDATED  .AND.   TRAILED   .AND.
LEDGERED)
           DELETE
         ENDIF
         SKIP
        ENDDO
```

```
        PACK
        @ 11,1 SAY 'Deletions complete.   '
       ENDIF && choice
       echoice='N'
       @ 11,1 SAY 'Delete old records from the payment file?  Y/N
' GET echoice FUNCTION '!'
       READ
       IF echoice='Y'
        @ 11,1 SAY SPACE(50)
        @ 11,1 SAY 'Deleting now..... '
        SELECT 10
        ZAP
        @ 11,1 SAY 'Deletions complete.'
       ENDIF && choice
       USE

     CASE choice='X'
      LOOP

     ENDCASE
    ENDDO  && choice<>'x'

    RETURN
```

```
*************************************************************
*                                                           *
*                      MAILLIST.PRG                         *
*                                                           *
*                            by                             *
*                                                           *
*                      Geoff Minshull                       *
*                                                           *
*                                                           *
*©copyright Geoff Minshull, 1987                            *
*************************************************************

*
*  Called from: REPORTS
*

*
*  This program prints out a mailing list of suppliers or
* customers. Can print all; search by name or town; those
* where balance>credit limit.
*

CLEAR

@ 1,40-(LEN(comname)*.5) SAY comname
@ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
bar='mailing list menu'
SET COLOR TO &col2/&col1
@ 24,0 SAY SPACE(80)
@ 24,1 SAY '<F1> for help'
@ 24,40-(LEN(bar)*.5) SAY bar
@ 24,71 SAY DATE()
SET COLOR TO &col1/&col2

menuck=.T.
del=.F.  && pack only if deletions made

SELECT 10
USE SUPPLIER

lsel=1              && selection number
lfile='SMAIL'      && name of label file
lfield=SPACE(5)    && to get correct field name for each file
lname=SPACE(20)    && name to search for
ltown=SPACE(20)    && town to search for
ptr=.F.            && printer control
tail=SPACE(45)     && command tail
pall=.F.           && print all indicator
fvalid=.f.         && check file selected
avalid=.f.         && check addresses selected

* column location of menu items in variable line

STORE '0113253240485867' TO locline
```

```
* length of items in locline

STORE '0808030404060504' TO lenline

STORE  'Supplier     Customer      All     Name      Town      Credit
Print    Exit' TO line
STORE  'Use  supplier file                                          '  TO
lmsg1
STORE  'Use  customer file                                          '  TO
lmsg2
STORE  'Select  all addresses                                       '  TO
lmsg3
STORE  'Select  addresses by full or part shortname                 '  TO
lmsg4
STORE  'Select  addresses by full or part town                      '  TO
lmsg5
STORE  'Select  addresses where balance greater than  credit'  TO
lmsg6
STORE  'Send  to screen or printer                                  '  TO
lmsg7
STORE  'Return  to previous menu                                    '  TO
lmsg8

DO WHILE menuck

 @ 4,1 SAY line

 SET COLOR TO &col2/&col1
 col=VAL(SUBSTR(locline,lsel*2-1,2))
 hilite=SUBSTR(line,VAL(SUBSTR(locline,lsel*2-1,2)),;
                    VAL(SUBSTR(lenline,lsel*2-1,2)))
 @ 4,col SAY hilite
 SET COLOR TO &col1/&col2
 STORE 'lmsg'+STR(lsel,1) TO lmsgnum
 @ 5,1 SAY &lmsgnum

  l=0

* routine to deal with dbaselll bug - not trapping leftarrow

  X=CHR(200)
  DO WHILE X=CHR(200)
   CALL INKEY WITH X
  ENDDO
  l=ASC(X)

  DO CASE

   CASE l=28
    DO HMAIL

   CASE l=1
    lsel=1

   CASE l=6
```

```
 lsel=8

CASE l=4  && right arrow
 IF lsel=8
   lsel=1
  ELSE
   lsel=lsel+1
 ENDIF

CASE l=19  && left arrow
 IF lsel=1
   lsel=8
  ELSE
   lsel=lsel-1
 ENDIF

CASE l=13  && return

 DO CASE

  CASE lsel=1
   fvalid=.t.
   SELECT SUPPLIER
   lfile='SMAIL'
   @ 10,1 SAY 'Supplier file chosen'

  CASE lsel=2
   fvalid=.t.
   SELECT CUSTOMER
   lfile='CMAIL'
   @ 10,1 SAY 'Customer file chosen'

  CASE lsel=3
   avalid=.t.
   lfield=' '
   lname=SPACE(20)
   Ltown=SPACE(20)
   @       11,1       SAY       'Print       all       record

   pall=.T.

  CASE lsel=4
   avalid=.t.
   pall=.F.
   lname=SPACE(20)
   DO WHILE lname=SPACE(20)
    @ 11,1 SAY 'Enter name: ' GET lname FUNCTION '!'
    READ
   ENDDO
   IF lfile='SMAIL'  && to deal with different field names
    lname='SSHORTNAME='+"'"+RTRIM(lname)+"'"
    ELSE
    lname='SHORTNAME='+"'"+RTRIM(lname)+"'"
   ENDIF
   ltown=SPACE(20)  && because only one selection possible
```

```
      CASE lsel=5
       avalid=.t.
       pall=.F.
       ltown=SPACE(20)
       DO WHILE ltown=SPACE(20)
        @ 11,1 SAY 'Enter town: ' GET ltown FUNCTION '!'
        READ
       ENDDO
       IF lfile='SMAIL'
        ltown='STOWN='+"'"+RTRIM(ltown)+"'"
        ELSE
        ltown='TOWN='+"'"+RTRIM(ltown)+"'"
       ENDIF
       lname=SPACE(20)

      CASE lsel=6
       avalid=.t.
       @ 11,1 SAY 'Print if balance exceeds credit limit'
       IF lfile='SMAIL'
        ltown='SBALANCE>SCREDLIM'  && use ltown to  avoid  another
variable
        ELSE
        ltown='BALANCE>CREDLIM'
       ENDIF

      CASE lsel=7
       CLEAR
       reply=' '
       DO WHILE reply<>'X'
        IF (.NOT. avalid) .OR. (.NOT. fvalid)
         @ 14,15 SAY 'You must choose a FILE and a SELECT option'
         READ
         @ 14,15 SAY '                                                '
         reply='X'
         LOOP
        ENDIF
        SET COLOR TO &col2/&col1
        @ 24,0 SAY SPACE(80)
        @ 24,1 SAY '<F1> for help'
        @ 24,40-(LEN(bar)*.5) SAY bar
        @ 24,71 SAY DATE()
        SET COLOR TO &col1/&col2
        reply=' '
        @ 3,1 SAY 'You can send them to the Screen - S '
        @ 4,1 SAY '                 or to the Printer - P '
        @ 5,1 SAY '                 or eXit           - X '
        @ 3,40 SAY 'Enter your choice: ' GET reply FUNCTION '!'
        READ
        IF READKEY()=36
         DO HMAIL
         LOOP
        ENDIF
        IF AT(reply,'PSX')=0
         LOOP
```

```
        ENDIF
        IF reply='X'
         LOOP
        ENDIF
       CLEAR
       IF pall
        tail=lfile
       ELSE
        tail=lfile+' FOR '+RTRIM(ltown)+RTRIM(lname)
       ENDIF
       IF reply='P'
        @ 18,1 SAY 'Press any key when printer ready          '
        WAIT ''
        tail=tail+' '+'TO PRINT'
       ENDIF
       LABEL FORM &tail
       WAIT
       CLEAR
      ENDDO

       CLEAR
       @ 1,40-(LEN(comname)*.5) SAY comname
       @ 2,40-(LEN(comname)*.5) SAY REPLICATE ('-',LEN(comname))
       SET COLOR TO &col2/&col1
       @ 24,0 SAY SPACE(80)
       @ 24,1 SAY '<F1> for help'
       @ 24,40-(LEN(bar)*.5) SAY bar
       @ 24,71 SAY DATE()
       SET COLOR TO &col1/&col2

      CASE lsel=8
       menuck=.F.

     ENDCASE

    OTHERWISE
     ? CHR(7)

   ENDCASE

ENDDO && menuck

menuck=.T.

SELECT 10
USE  && close file

RETURN
```