1987

LIES library information and enquiry system

Kerrie Barker  
*University of Wollongong, uow@barker.edu.au*

Helen Bow  
*University of Wollongong, uow@bow.edu.au*

Vicki Grossmann  
*University of Wollongong*

Andrew Byrnes  
*University of Wollongong, uow_byrnes@uow.edu.au*

Allan Hird  
*University of Wollongong*

---

**Recommended Citation**

Barker, Kerrie; Bow, Helen; Grossmann, Vicki; Byrnes, Andrew; and Hird, Allan, LIES library information and enquiry system, Department of Computing Science, University of Wollongong, Working Paper 87-3, 1987, 128p.  
http://ro.uow.edu.au/compsciwp/29

---

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
LIES is a Library Information and Enquiry System that provides a user friendly interface and fast access to both the catalogue and borrower information stored in the system.

The user friendly interfaces are provided through the use of tailored menus for the different users of the system:

* the borrowers
* the Librarian
* the Library Manager

and screen based on line help for all functions.
INDEX

PART 1. Technical Report


PART 4. User Tutorial
LIES
LIBRARY INFORMATION AND ENQUIRY SYSTEM

Kerrie Barker
Helen Bow
Vicki Grossmann
Andrew Byrnes
Allan Hird

Submitted in partial fulfillment of the requirements of CSCI 321 Software Project at the University of Wollongong in 1986
ABSTRACT

This document is the technical report for an interactive screen based maintenance, circulation and information retrieval system for a small Library run on the Unix Operating System on the Computing Science Pyramid computers.
TABLE OF CONTENTS

INTRODUCTION

RUNNING THE SYSTEM

OVERVIEW

1. DEFINITIONS

2. FILE SYSTEM
   2.1 Design
   2.2 Updates

3. USER INTERFACE
   3.1 Screen Handling
   3.2 Input Data Collection
   3.3 Security
   3.4 On-Line Assistance

4. MAINTENANCE FUNCTIONS
   Overview
   4.1 L0201.c Maintain Catalogue Information
   4.2 L0202.c Maintain Borrower Information
   4.3 L0203.c Maintain Subject Information

5. RETRIEVAL FUNCTIONS
   5.1 L04.c Catalogue Information Retrieval
   5.2 L0303.c Borrower Information Retrieval

6. CIRCULATION FUNCTIONS
   6.1 L0301.c Borrow An Item
   6.2 L0302.c Return an Item

7. REPORTING FUNCTIONS
8. GLOBAL FUNCTIONS

9. MAINTENANCE

10. ENHANCEMENTS

11. APPENDIX

11.1 File Structures

11.2 System Constants

11.3 Standard Screen

11.4 File Names
**INTRODUCTION**

LIES is a Library Information and Enquiry System that provides a user friendly interface and fast access to both the catalogue and borrower information stored in the system.

This manual is intended to be read in conjunction with the L.I.E.S System Design document.

**RUNNING THE SYSTEM**

To start the L.I.E.S system type the command

```
LIES
```

at the UNIX system prompt. This command invokes the menu driver program of the system (L01.x). All menus are contained in this module. All menu selections invoke the related programs via UNIX system calls. The separation of the individual functions into separate programs invoked by UNIX system calls as opposed to running the whole system as one (1) program means we are able to duplicate variable and procedure names within the system thus making programming easier.
OVERVIEW

The L.I.E.S system is designed for the following users:
* One Library Manager
* One Librarian
* Any number of borrowers

The Library Manager and Librarian can both perform functions which update files. Since no file locking is provided these functions must be single user to avoid corruption of files or a possible system crash. For the same reason the Library Manager and Librarian cannot both be using the system at the same time. Since borrowers only read access the Catalogue file any number of them can run at the same time, along with the Librarian or the Library manager.

This system allows the users of the system to interactively

* Perform file maintenance functions,
  L0201.c Maintain Catalogue
  L0202.c Maintain Borrower
  L0203.c Maintain Subjects

* Perform Retrieval (or searching) functions,
  L04.c Catalogue Information Retrieval
  L0303.c Borrower Information Retrieval

* Perform Circulation functions, and
  L0301.c Borrow an Item
  L0302.c Return an Item

* Submit reports,
  L020501.c Catalogue Report
  L020502.c Borrower Report
  L020503.c On-Loan Report
  L020503.c Overdue Report
  L030303.c Borrower Search Results
  L0304.c Subject Definitions
  L040301.c Catalogue Search Results
2. **FILE SYSTEMS**

2.1 **DESIGN**

The L.I.E.S system utilises thirteen (13) files. The files are divided into four (4) categories: data files, transaction files, index files and memory resident data file.

The data files for the system are

- Catalogue File
- Borrower File
- On-Loan File

Each data file contains fixed length data records in ascending order of the unique primary key. Access to the files can be both sequential and direct. The primary key for each data file is as follows:

- Catalogue: Item_number
- Borrower: Borrower_number
- On-Loan: Item_number + Copy_number

The index files for the system are

- Author Index File
- Title Index File
- Subject Index File
- Borrower Name Index File

and provide an inverted file design approach to indexing. The index files are sorted on a non-unique key. Each index record will contain a record position of the corresponding data record in the data files. The data file can then be accessed directly. The primary key for each index file is as follows:

- Author Index: Data_author.surname
- Title Index: Title_name
- Subject Index: Subject_identity
- Borrower Name: Data_borrower.surname

The transaction files for the system are

- Catalogue Transaction File
- Borrower Transaction File
- Daily (On-Loan) Transaction File
- Subject Transaction File

and will contain all the transactions for the respective data file that have been accrued since the last update of the data file. The records will be in date/time order.
The memory resident data file is the Subject Reference File.

This file is a relative file which initially resides on disk. Had our system been treated as one large program this file at system startup would have been read into main memory and treated as an array. By implementing the system as many independant programs this is no longer possible and so each time a program which requires to use this array (L04.c, L0201.c, L0203.c) is invoked it must call L0101.c to read the file into memory. The subject table has been defined as an N-way tree structure, but implemented as a static array for the following reasons:
- subjects are pre-defined and, as such, the relationships between subjects are pre-defined.
- the subject heirachy does not change often, and as such a dynamic data structure was considered unnecessary.
- a subject search is usually linear. In the worst case accessing an array linearly is quicker than accessing a dynamic data structure.

All file structure definitions (App 11.1) are in separate files. This allows for easy maintenance since a change to any file structure need only be done in one place and the appropriate modules recompiled.
2.2 UPDATES

The L.I.E.S system processes most updates in BATCH mode. The major file update sequences are

Catalogue Updates.
Borrower Updates.
On Loan Updates.

These updates cannot be run while the system is in use and so must be done overnight.

These updates are each SHELL procedures which invoke a series of C programs and UNIX system commands to perform the update sequence.

The Catalogue Update sequence (L020401.sh) is a sequence of eight (8) procedures that will update the Catalogue Data File from the transaction files and recreate all the index files. The sequence steps are:

1. Copy all files to a backup copy for safe keeping.
2. Sort the Catalogue Transaction file by item_number and date/time of transaction. Select only the latest transaction for each item_number and rewrite the transaction file.
3. Use a standard merge algorithm to merge the Catalogue Transaction file with the Catalogue Data File.
4. Update the Catalogue Data File with any changes to the subjects that appear on the Subject Transaction File.
5. Create the Catalogue Index Files (Author, Title, Subject) by processing the Catalogue Data File.
6. Sort the Catalogue Index Files.
7. For each subject in the Subject Reference File, update the record offset pointer into the Subject Index File. (This will save binary searching the Subject Index File.)
8. Copy all the new files into position for use in the system.

The Borrower Update sequence (L020402.sh) is a sequence of six (6) procedures that will update the Borrower Data File from the transaction file and recreate the Borrower name Index File. The sequence steps are:

1. Copy all files to a backup copy for safe keeping.
2. Sort the Borrower Transaction file by borrower number and date/time of transaction. Select only the latest transaction for each borrower and rewrite the transaction file.
3. Use a standard merge algorithm to merge the Borrower Transaction File with the Borrower Data File.
4. Create the Borrower Name Index File by processing the Borrower Data File.
5. Sort the Borrower Name Index File.
6. Copy all the new files into position for use in the system.
The On Loan Update sequence (L020403.sh) is a sequence of four (4) procedures that will update the On Loan data file from the transaction file. The sequence steps are:

1. Copy all files to a backup copy for safe keeping.
2. Sort the On Loan (daily) Transaction File by item_number / copy_number and date/time of transaction. Select only the latest transaction for each item_number/copy_number and rewrite the On Loan (daily) Transaction File.
3. Use a standard merge algorithm to merge the On Loan Transaction File with the On Loan Data File and carefully detect any exceptions.
4. Copy all the new files into position for use in the system.

L020404.sh runs all these updates processes.

To check that these updates have run and completed successfully the library manager must exit from the LIES system and examine a UNIX file. When an update is submitted informative messages are piped from the shell procedure into a UNIX file of the name "the prefix of the shell file".lis. For instance the catalogue updates go to L020401.lis.

A message will appear for each of the steps signifying it completed successfully. When the entire procedure has completed successfully the message "filename" update completed successfully will be the last message displayed. If this does not appear then the last step completed can be detected from the messages and the user will know the problem occurred somewhere in the following step. For more information the file "the prefix of the shell file".err will contain further messages pinpointing where the problem occurred.

If any update fails to complete all steps successfully it will the old data can be recovered by copying the backup files back into the original data files. All backup files have an extension of .OLD.
3. USER INTERFACE

The L.I.E.S user interface is entirely screen based. All user interaction with the system is via menus or screen functions.

3.1 SCREEN HANDLING

Screen handling is controlled by the CURSES package. It permits the program to position text on the screen, to position the cursor appropriately for the input of data fields, and to allow the movement of output fields, messages and highlighted fields to the screen.

To maintain consistency and ease of operation, a standard screen has been used in the L.I.E.S system. (APP 11.3). A set of standard control keys are also used throughout the system. These are:

- CNTL-A : Help
- CNTL-C : Cancel Key
- CNTL-E : Function Terminator
- CNTL-P : Print Results Report
- TAB : Field Terminator
- RETURN : Submission Key

Corresponding to all input screens is a data structure - SCREEN_CURRENT, where the input from the screen is stored for subsequent processing. All input to the screen is placed initially in character fields. Data which is required to be numeric is then checked whether numeric and converted to its corresponding numeric field using EDIT_CONV_INT. Each screen containing numeric data will also have a numeric screen data structure corresponding to its string/character data structure. Integer fields are stored in this structure to be passed to the processing phase of the function.

FIELD INPUT

All input of fields is handled by a common function - GETFIELD. This function is invoked by:

code_return = GETFIELD (field, length)
field = current screen field or data item.
length = length of field in characters.
code_return - GETFIELD will return a code to indicate the type of control key used to terminate the input field. It may be set to the values
CD_RETURN, or
CD_CANCEL, these are initialised in the constants file - constants.h. This field may then be checked by the calling function to see how it should proceed.

field - upon return to the calling program it will contain what has been input to the screen by the user. If no valid character input was entered or the field was blanked out or cleared it will contain a null character.
All characters typed on the screen are filtered by GETFIELD. This is achieved by setting the terminal device in RAW mode. Thus, rather than characters such as QUIT, interrupt and suspend generating a signal the characters are passed through uninterpreted. This allows the function total control over what the user types (eg. the user cannot break out of the system).

All characters are converted to upper case. This bypasses any ambiguity associated with searching on upper or lower case fields.

When a field is terminated GETFIELD returns the characters typed into that field to the calling function. GETFIELD will be terminated in one of 5 ways:
- detection of TAB
  terminates the field and returns to the calling function.

- detection of RETURN
  returns to the calling function and sets the global field code_return to CD_RETURN so the calling function can recognise that screen input has been terminated.

- detection of CNTL-E
  resets the terminals to normal operating mode and exits the current process back to the process that invoked it. Uses the UNIX system command - exit.

- When the field is full
  returns to the calling function.

- detection of CNTL-C
  returns to the calling function and sets code_return to CD_CANCEL so the calling function can recognise that screen input has been cancelled.

FIELD EDITING

GETFIELD allows various edits to be performed on the field by the user. The CURSES package allows the following keys to be recognised and GETFIELD determines what function is to be performed when the key is pressed:
- BACKSPACE: delete last character typed
- LEFT ARROW: as for backspace
- RIGHT ARROW: moves the cursor right until the last character of the field is reached
- DEL: delete current character
- HOME: move to beginning of field
- ESC: clear to end of field
- CLEAR: erase entire field and place the cursor at the beginning of the field.

It is possible to recognise the DEL and ESC keys because the device is set in RAW mode.
VALIDITY CHECKS

GETFIELD checks for a set of valid characters. These include any single character which can be typed from the keyboard, and the help (CNTL-A), exit (CNTL-E) and cancel (CNTL-C) control keys. Leading blanks are ignored for any field longer than one character. Trailing blanks are trimmed before the field is returned. An 'INVALID KEY' message is issued if any other key is hit.
3.2 INPUT DATA COLLECTION

The collection of input is concerned with collecting the user input from the function's input screen and placing it in the screen data structure to be passed on for processing. All screen functions perform the same basic processing in this area. Screen function.

Once a screen function is invoked the input screen for that function is displayed to the user. On all the functions the cursor will initially be positioned at the first input field for that screen, awaiting user input. Simple field validation is handled by placing a loop around the retrieval and consequent validation of the field. While the validation requirements for that field are not met or the cancel key or function termination keys are not hit then the cursor will be positioned back at that field and a suitable error message displayed, until it is valid. The field terminator (TAB key) therefore, cannot be used to pass by a field until the data input into that field is valid.

Once the inputted field is valid the cursor is placed at the next field which requires input, moving from left to right and down the screen. Once the last field has been input, the cursor is placed back at the first input field. This wrap around effect is achieved by placing the collection of the entire screen data within a loop. This loop is terminated when the use of the submission key is detected. When the cancel key is detected as input the function will break from any validation loop it may be currently in, clear all the input fields and return to the top of the loop ready to accept the first input field.

The highlevel algorithm for this phase is

while
  position cursor at input field1
  while input field1 not valid
    GETFIELD input field1
    if cancel key detected
      break from loop
    validate input field1
    if not valid
      return error message
    postion cursor at input field1
  end
  if submission key detected
    break from loop
  if cancel key detected
    reset screen
    continue loop

position cursor at input field2
GETFIELD input field2
if cancel key detected
  reset screen
  continue loop
if submission key detected
  break from loop
end
3.3 SECURITY

Access to the Librarian and Manager functions is restricted in order to prevent misuse by unauthorised users. Two methods of protection have been employed:
- "hidden" access method.
- password protection.

HIDDEN ACCESS

To deter the curious user, instructions on how to access the restricted functions are not given in the L.I.E.S logo screen.

PASSWORD PROTECTION

The password for the Manager and Librarian functions can be different. Two programs:

L0102.o    Check_manager_authority
L0103.o    Check_librarian_authority

are used to check security. The current password is stored in an ASCII file. The password entered on the Enter Password screen is read by the function GET-PASSWORD which operates similarly to GETFIELD (see later explanation) except any characters entered are not echoed. Another two modules:

L0206.o    Change_manager_password
L0305.o    Change_librarian_password

are used to change the password after accessing the required function.
3.4 ON-LINE ASSISTANCE

L.I.E.S offers the user assistance in 2 ways:
- Full screen help
- Informative messages

HELP SCREENS

Each screen has an associated help screen which can be invoked by pressing CNTL-A. This invokes the function HELPINFO which displays a screen structure - HELPWIN - over the current window.

The help screens are stored in ASCII files with one file per screen function. The files are up to 80 characters wide and can be as long as necessary. The text can be paged through one screen at a time. A block of seventeen lines makes one page. The first seventeen lines are read from the help file and displayed. Pressing the space bar causes the next 17 lines to be read and displayed. At end of file, pressing RETURN returns control to the calling function and its screen is redisplayed.

Each screen is given a unique identity which is defined in the system constants file. This identity allows the help function to determine which help file to read. SCREEN_IDENTITY is a global variable available to all procedures called by the screen function it must be set to the identity for the help required at the time.

For a list of help file names see APP 11.5. The names have an extension of .HLP and a file name associated with the unique screen identity.

INFORMATIVE MESSAGES

All error messages generated by L.I.E.S functions are displayed on line 2 of the screen and highlighted in reverse video. An alarm is run (ie beep) when an error occurs. Any informative messages for the user are displayed on line 3 in a similar fashion.
4. MAINTENANCE FUNCTIONS

OVERVIEW

The library manager is responsible for performing the updates on the data files within the LIES system. In order to do this he/she is provided with a number of maintenance programs and to verify the files he/she has the option of producing reports.

Three maintenance functions are provided in the LIES system:

- MAINTAIN CATALOGUE INFORMATION
- MAINTAIN BORROWER INFORMATION
- MAINTAIN SUBJECT REFERENCE DATA

For consistency the functions share common fields and processing. The TRANSACTION_CODE field on the screens allows the user to nominate the process he wishes to perform. The alternatives are:

- 'D' - for DISPLAY
- 'C' - for CREATE
- 'D' - for DELETE
- 'U' - for UPDATE

If a DISPLAY is required, only the screen key field must be entered, i.e., ITEM_NO for catalogue displays, BORROWER_IDENTITY for borrower displays and SUBJECT_IDENTITY for subject reference displays. In the case of a DISPLAY, the programs restrict movement of the cursor to the TRANSACTION_CODE field or screen key field only.

In CREATE mode, the user is able to enter all fields on the screen. In order to UPDATE records the user must first DISPLAY the record. The process was designed in this way so as to provide an extra safety mechanism to the user. By displaying the record first he can be certain at a glance that he is updating the correct data. Also, altering the fields is made easier as he can see what the record currently looks like and simply type over the data in question. In this case batch processing is simplified as well since in an UPDATE situation the full record is written to the transaction file not just the fields that were altered.

If the user types in a 'U' for UPDATE without having done a successful DISPLAY the cursor will remain at the TRANSACTION_CODE field and display an error message. Once a successful DISPLAY is achieved the user is able to alter all fields on the screen except for the screen key field which is protected by the function.

In order to DELETE records the user must first DISPLAY the record. As described above, this is to provide safety mechanisms to the users. It is hoped that this will reduce erroneous deletes by the user.

For a DISPLAY, a binary search using the appropriate BINSEARCH function is performed on the data file. Since the TRANSACTION file is not sorted and we require to locate the most up to date record for the item, a sequential search is conducted on the TRANSACTION file. Thus if a record is found on the TRANSACTION file then it is displayed, otherwise if a record is found on the CATALOGUE file then it is displayed else an error message is returned.
For a CREATE or an UPDATE or a DELETE a record is written to the TRANSACTION file for batch updating. Refer structure of the TRANSACTION file (Appendix 11.1). A time stamp is put on the record as it is required by the batch updating program. The data entered by the user or the deleted record remains on the screen and an informatory message is displayed. This is done so that in cases where the user has made an error he may notice it straight away. For a DELETE on a BORROWER or a CATALOGUE record if the record is in some way linked to an ONLOAN record the function will not permit the deletion in order to maintain file integrity. Similarly a SUBJECT record may not be deleted if it has children.
4.1 L0201.c  MAINTAIN CATALOGUE INFORMATION

SCREEN DATA INTEGRITY

TRANSACTION_CODE : Must be a ' ', 'C', 'D' or 'U'.

ITEM_NUMBER : Must be numeric and non-zero. The function EDIT_CONV_INT is used for verification. Also if TRANSACTION_CODE = 'C' the program validates if the ITEM_NUMBER entered already exists by performing a binary search on the CATALOGUE file and a sequential search on the TRANSACTION file.

MEDIUM : Must be one of the following
- default value of 'B' is assumed.
- 'A' - Audio
- 'B' - Book
- 'J' - Journal
- 'M' - Manual
- 'S' - Software
- 'V' - Video
- 'O' - Other (miscellaneous)

SURNAME 1 : Compulsory.

FIRST GIVEN NAME :

SECOND GIVEN NAME : Can only be entered if the first given name is entered. If the first given name is blanked out then this field is blanked out by the function to maintain file integrity.

SURNAME 2 : If blanked out and given names exist for it then to maintain file integrity the first names are blanked out by the function.

FIRST GIVEN NAME : Can only be entered if the second surname is entered.

SECOND GIVEN NAME : Can only be entered if the first given name is entered. If the first given name is blanked out then this field is blanked out by the function to maintain file integrity.

TITLE : Compulsory.

DATE_PUBLISHED : Must be numeric and less than the current year. Function EDIT_CONV_INT is used for numeric validation and the function CURRENT_YEAR is used to for comparison.

PUBLISHER NAME : Since the length of this field is greater than the screen width it had to be divided into two screen fields. The first of the two publisher name fields is compulsory. Only if the user enters the whole first field can he then enter the second publisher field.
EDITION NUMBER: Compulsory. Must be numeric and non-zero. The function EDIT_CONV_INT is used for validation.

DEWEY NUMBER: Compulsory.

NUMBER COPIES: Compulsory. Must be numeric and non-zero. The function EDIT_CONV_INT is used for validation.

SUBJECTS: At least one must be entered. The global function VAL_SUBJ_TAB is used to verify if the subject is present in the predefined subject list. If the Subjects entered cannot have a parent-child relationship so the global function CHECK_PARENT is used to verify this. If the user blanks out a subject the function will shuffle remaining subjects to the left.

VERIFICATION

This involves checking that all compulsory fields have been inputted and that the combination of subjects entered is valid. This is required in certain cases where only part of the data has been entered or altered. For instance, during a CREATE, if the user presses RETURN after entering the title of an item, the program must ascertain if the remaining fields had been entered as well.
SCREEN DATA INTEGRITY

TRANSACTION CODE : Must be ' ', 'C', 'U' or 'D'.

BORROWER_ID : Must be numeric and non-zero. The function
EDIT_CONV_INT is used for verification. Also if
TRANSACTION_CODE = 'C' the program validates if
the BORROWER_ID entered already exists by
performing a binary search on the BORROWER file
and a sequential search on the TRANSACTION file.

SURNAME : Compulsory.

FIRST GIVEN NAME : 

SECOND GIVEN NAME: Can only be entered if first given name is
entered.

STREET NUMBER : Compulsory.

STREET NAME : Compulsory.

SUBURB : Compulsory.

POSTCODE : Compulsory.

TELEPHONE NUMBER :

VERIFICATION

This involves checking that all compulsory fields have been inputted,
as described in MAINTAIN CATALOGUE INFORMATION above.
SCREEN DATA INTEGRITY

SUBJECT : Unless in CREATE mode the subject entered must be present in the predefined subject dictionary. If in CREATE mode the program checks if the subject entered already exists.

PARENT : The parent can only be entered during CREATE mode and the subject must be in the predefined subject dictionary.

DESCRIPTION : Not compulsory. Can be entered during CREATE mode or altered during UPDATE mode.

For a DISPLAY, the parent and description for the subject entered is retrieved from the subject table.

For a CREATE, the function determines whether the subject created is the only one for that particular parent by checking if CHILDPOSITION is NULL. This is required in order to update the respective pointers in the subject tree. The new subject is also added to the subject table.

For an UPDATE, the user is able to change the spelling of a subject by first displaying the subject and then updating it. The function will then change the old subject to the new subject. A record is also written to the SUBJECT TRANSACTION file for batch updating on the CATALOGUE file.

For a DELETE, the function will delete that subject from the subject tree, update any pointers that were affected and remove the subject from the subject table.

When the function is exited, the Subject Reference Table on disc must be rewritten with the new version from memory, so that next time a function sets up the array from the file on disc they will be getting the updated tree. To achieve this GETFIELD cannot be used to read in the SUBJECTS from the screen since on detection of the exit key it will perform the exit to the invoking process without returning control to L0203.c. A new procedure GET_SUBJECT is used (see description in section 6. of this report). GET_SUBJECT upon detection of the exit key, terminates the field and sets CODE_RETURN to CD_EXIT. L0203.c may then check this and upon exiting rewrite the file on disc.
5. RETRIEVAL FUNCTIONS

The Retrieval Functions enable the user to retrieve information about items or borrowers using a variety of non-unique search criteria.

There are two of these functions
i) Borrower Information Retrieval
ii) Catalogue Information Retrieval
5.1 L04.c CATALOGUE INFORMATION RETRIEVAL

OVERVIEW

This function enables the user to enter and effect a search on Catalogue information.

The function is divided into three (3) functionally independent modules. Namely -
1) Collect Search Criteria (L0401.c)
2) Process Search (L0402.c)
3) Display Results (L0403.c)

The mainline (L04.c) performs the input verification before calling Process Search.

L0401.c COLLECT SEARCH CRITERIA

SCREEN DATA INTEGRITY

AUTHOR

SURNAME

FIRST GIVEN NAME: Can only be entered if the surname is entered.
Blanked out by the function if the SURNAME blanked out by the user.

SECOND GIVEN NAME: Can only be entered if the first given name is entered. Blanked out by the function if the SURNAME or FIRST GIVEN NAME blanked out by the user.

TITLE

MEDIUM: Must be one of the following
' ' - all are searched for
'A' - Audio
'B' - Book
'J' - Journal
'M' - Manual
'S' - Software
'V' - Video
'O' - Other (miscellaneous)

DATE_PUBLISHED1: Must be numeric and less than the current year.
Function EDIT_CONV_INT is used for numeric validation and the function CURRENT_YEAR is used to for comparison. It must also be less than DATE_PUBLISHED2 if it is input.

DATE_PUBLISHED2: Must be numeric and less than the current year.
Function EDIT_CONV_INT is used for numeric validation and the function CURRENT_YEAR is used to for comparison. It must also be greater than DATE_PUBLISHED1 if it is input.
SUBJECTS

The function VAL_SUBJ_TAB is used to verify if the subject is present in the predefined subject list. If the user blanks out a subject the function will shuffle remaining subjects to the left if need be. The first subject may contain a '!' provided all other subjects are blank.

SELECTION

Only required during the ASSISTED SEARCH procedure. Must be required for submission and the display of a lower level. Must be less than or equal to the number of subjects displayed on the screen.

The Collect Input phase of this function varies from all other functions in that it provides an assisted subject selection facility to the user. This is contained in the procedure ASSISTED_SEARCH, which is called from the COLLECT_SEARCH_CRITERIA , if a '!' is detected in the first subject field of the screen, provided no other subjects have been entered.

A STACK is used to store the selections of the user. The STACK is set at five (5) elements which allows it to function on a Subject Reference Tree of six levels counting the root level. Currently this is ample, any increase in the number of levels thus will mean this STACK_SIZE must be increased. The standard POP, PUSH and INITIALISE algorithms are used to manipulate the stack.

Currently, the subject field is fifteen (15) characters, this enables us to display four (4) of them across the screen, on eight (8) lines and thus we may display thirty-two (32) subjects. This places a limitation of this amount (32) on the number of subjects which may occur on each branch of the tree. This amount at present is ample and we believe should remain so for the size of the library we envisage. If more than thirty-two (32) subjects on a branch is required then it is not an easy matter to display more. It either means a reschedule of the screen to supply more lines to display them on or a decrease in the size of the subject field to fit more on each line. If a need to increase the size of the subject field arises it should be noted that we may not be able to fit as many on the screen and so the limit on the number of subjects on each branch of the tree is decreased. Thus a trade off between subject field size and the number of subjects on each branch of the tree is required.

VERIFICATION (performed in L04.c)

Either the title, the author surname or at least one subject must be entered. Searching is only performed on the lowest child of each branch of the tree.

CHECK_SUBJECT

The global CHECK_PARENT is used to check whether one subject is a descendant of another and if so only the descendant will be written into a searchable subject structure, this is then checked against the third subject, and if one is a child then only it is written to this structure. If none are descendants then all are searched on. This is to avoid repetitive searching since a search performed for items on a subject also retrieves all the items for its descendants. When the search results for a subject and for its descendant are anded together only the matches for the descendant will result anyway. The procedure where this is performed (CHECK_SUBJECT) is complex but is well
commented and should so be easy to follow.
Since a combination of search criteria is treated as a conjunctive search, it is necessary that when the matches for one criterion are found, the position number of these matching records in the Catalogue File be stored to see whether they are also matches for the other criteria. This is achieved by placing the position number of the match in a BINARY SORT TREE along with a counter which is incremented each time that position number is seen to match one of the criteria. Once all the criteria have been searched on, the tree is traversed and only those which have counters equal to the number of search criteria are retrieved from the Catalogue file and written to the Output file. All tree accessing is achieved using recursive functions, REAL_TREE and REAL_TRAVERSE, the algorithms of which are the standard form. The tree is dynamic, each node allocated when required and thus as many subjects as required may be stored in it.

The processing for a Subject search differs because the situation exists where an item is classified under multiple subjects. For instance, if a search is performed on "DATA" and item "A" has been classified under the subjects "DATABASE" and "DATA STORAGE", two of the children of "DATA", it will have two occurrences on the Subject Index File which will both be retrieved as a possible match for "DATA". This means the count on the tree for this item will be incremented to two (2), even though we are only looking for those with a count of one (1) since we only have one search criteria. To overcome this, a separate function for placing subject matches in the tree (TREE_SUBJ) is utilized. It makes use of an array of three (3) flags (FLAG_SUBJECT) which are used in conjunction with the tree to signify whether that item has already been found for the searchable subject we are currently examining.

Straight forward read from Output file and display to the screen of three (3) records at a time. Scrolling up (the up arrow) is simply done by an LSEEK back through the file six (6) records, the next three (3) records are read and moved to the screen. The print key provides a hardcopy of the entire Output file by system call to a SHELL file which submits the print function (L040301.c).

Like the Catalogue record.
5.2 L0303.c BORROWER INFORMATION RETRIEVAL

OVERVIEW

The aim of this function is to provide quick retrieval of information on a borrower including their number, name, address and the items they have on loan and when they are due to be returned. The function is divided into four (4) functionally independent modules. Namely -

1) Collect Search Criteria (L030301.c)
2) Process Search     (L030302.c)
3) Display Results    (L030301.c)
4) Display Mult Res   (L030301.c)

The mainline (L0303.c) performs the verification of input before calling Process Search.

L030301.c COLLECT SEARCH CRITERIA

SCREEN DATA INTEGRITY

INPUT

BORROWER NUMBER: Must be numeric and non-zero. The function EDIT_CONV_INT is used for verification.
Only allowed to enter if BORROWER SURNAME is blank.

BORROWER NAME
SURNAME: Only allowed to enter if BORROWER NUMBER is blank.

GIVEN NAME (1): Only allowed to enter if BORROWER SURNAME non blank. If the SURNAME is blanked out by the user then the function will blank this field also.

GIVEN NAME (2): Only allowed to enter if GIVEN (1) non blank. If the SURNAME field or the GIVEN (1) field are blanked out by the user then this field will be blanked out by the function also.

OUTPUT

BORROWER ADDRESS
STREET NUMBER
STREET NAME
SUBURB
POSTCODE
ITEM DATA (10)
ITEM NUMBER
TITLE
DATE DUE BACK

VERIFICATION (performed in L0303.c)

Either the borrower number or borrower surname must be entered for the search.
When the BORROWER NUMBER is input the record is retrieved by doing an binsearch on the Borrower file. If the borrower has items on loan then for each item the Onloan file is binary searched. If the Onloan record for that item is not located then it is assumed that it was borrowed on that day and the DATE DUE BACK on the output data structure is made the current date plus seven days. If the record is located and the borrower identity on it is blank then the item has been returned today and so is not displayed. The Catalogue file is then binary searched, if the record is not located then a file error is assumed and appropriate action taken. The required data from all the records is written to the output record and the output record written to the output file.

When the BORROWER SURNAME is input then the index file must be accessed first and any matching records found here are then retrieved by doing an LSEEK on the Borrower file to the offset on the index. If only one record is found matching then processing continues as above for the BORROWER NUMBER. However, if more than one match is located then for each match only the data from the borrower record is written to the output structure and subsequently to the output file. This avoids extra file accessing on the Onloan and Catalogue files when the user is unsure which borrower they actually require the data for. Supplying the borrower data enables them to decide which borrower they actually wanted and then resubmit the search with that BORROWER NUMBER for that borrower.

If one record is retrieved then this Output phase reads the output record from the output file and moves it into the screen output structure and displays it on the screen. A print is provided calling the print function (L030303.c).

If more than one record has been retrieved then the output file is treated in the same way as in CATALOGUE INFORMATION RETREIVAL described previously, except one record is displayed at a time. There is no print key provided in this area. The submission key is accepted, and if it is input then the PROCESS_SEARCH is recalled this time with the BORROWER NUMBER.
6. **CIRCULATION FUNCTIONS**

The librarian is responsible for performing the daily duties of the library. In order to do this he/she is provided with two circulation programs these being **BORROW AN ITEM** and **RETURN AN ITEM**.
6.1 L0301.c  BORROW AN ITEM

This process allows the librarian to enter information about an item being borrowed, to be kept as a record of the transaction.

SCREEN DATA INTEGRITY

BORROWER_ID  : Compulsory. Must be numeric and non-zero.
ITEM_NO       : Compulsory. Must be numeric and non-zero.
COPY_NO       : Compulsory. Must be numeric and non-zero.

VERIFICATION

Checks borrower exists and has less than the maximum number of books on loan (10).
Checks the item exists and is not presently on loan. If it is on loan then it is up to the librarian to investigate the discrepancy.

PROCESSING

To ensure up to date information on the CATALOGUE file and BORROWER file on-line updating is performed.

Number copies borrowed on the Catalogue file is updated immediately. This is important as users requesting catalogue data by using the RETRIEVE CATALOGUE INFORMATION function will know exactly how many copies are available in the library.

Items on loan on the Borrower file is also updated immediately in the same manner. Number items on loan is incremented and the item borrowed is inserted in the list of items borrowed by that borrower. This is so that borrower information displayed using the RETRIEVE BORROWER INFORMATION function is accurate and that if the situation arises that a borrower has the maximum books on loan he/she is able to return a book and borrow another book on the same day.

A record of the transaction is written to the DAILY TRANSACTION file for batch processing. During batch processing any integrity errors are reported.
6.2 L0302.c RETURN AN ITEM

This process allows the librarian to enter information about an item being returned.

SCREEN DATA INTEGRITY

ITEM_NO : Compulsory. Must be numeric and non-zero.
COPY_NO : Compulsory. Must be numeric and non-zero.

VERIFICATION

Check to see if the item is presently on loan.
Check the item exists.
Check the borrower reported to have the book exists and has that book on loan.

PROCESSING

Up to date information is also achieved in this function as described in the BORROW AN ITEM function.

Number copies borrowed on the CATALOGUE file is decremented.

Number items on loan on the Borrower file is decremented and the item returned is removed from the list of items borrowed by that borrower. The function will shuffle any remaining items to the top of the list if need be.

The ON LOAN file is also updated immediately. The borrower identity for the record is blanked out so to indicate that the item was returned today.

A record of the transaction is written to the DAILY TRANSACTION file for batch processing. Any integrity errors are reported by batch processing.
7. REPORTING FUNCTIONS

The L.I.E.S systems provides the facility to generate seven (7) reports. The CATALOGUE, BORROWER, ON-LOAN and OVERDUE reports are submitted via the MANAGEMENT REPORTS menu. The SUBJECT DEFINITIONS report is submitted from the LIBRARIAN FUNCTIONS menu. The remaining two reports can be generated by the print key ( CNTL-P ) on the display catalogue results and display borrower data screens.

When a report request is issued a batch program is submitted and its output is piped to the printer. To supress the UNIX system message which is issued when a print request is made the -s option is used on the lp command. An informatory message is displayed to tell the user the report has been submitted and he/she can continue working.
8. GLOBAL FUNCTIONS

The L.I.E.S system has a number of functions used in several programs. Two of these (GETFIELD & HELPINFO) have already been discussed in detail.

BINARY SEARCH

There are six (6) binary search functions. Each searches a different file. These files are:
- Catalogue
- Borrower
- On-Loan
- Author Index
- Title Index
- Borrower Name Index

Usage:

```c
CAT_BINSEARCH( item_number )
item_number = unique key to Catalogue File
```

```c
BORROWER_BINSEARCH( identity_borrower )
identity_borrower = unique key to Borrower File
```

```c
ON_LOAN_BINSEARCH( item_number, copy_number )
item_number
copy_number = combine to make a unique key of the On-Loan file.
```

```c
AUTHOR_BINSEARCH( surname )
surname = non-unique key to Author Index File
```

```c
TITLE_BINSEARCH( title )
title = non-unique key to Title Index File
```

```c
NAME_BINSEARCH( surname )
surname = non-unique key to Borrower Name Index File
```

Description:

Each of these functions performs a binary search for the data passed, on a specific file. In cases where the data occurs more than once the first occurrence is returned. If the data is not found the functions all return the value -1. If the data is found the record is stored in a data structure defined in the calling function. The CAT_BINSEARCH, BORROWER_BINSEARCH and ON_LOAN_BINSEARCH functions use the file descriptor from the calling function and leave the file descriptor at the current position in the file. This position is used in programs which perform direct updates on files (e.g., borrow an item). The program can return to the correct position in the file to rewrite the record. The remaining three return the number of characters last read.
CALC_DATE

Usage:

CALC_DATE( string )

string = a 12 character string in which current date and time is returned

Description:

This function returns today's date and time in the form DD MMM YY HH:SS. The current date value is obtained using the UNIX system call TIME() which returns a long integer value for the current date and time. This value is converted to a 26 character string using the C subroutine CTIME(). The string returned is substringed to the 12 characters required.

CURRENT_YEAR

Usage:

n = CURRENT_YEAR()

n = a 4 digit integer containing the value of the current year.

Description:

This function returns the year in the form YYYY. The current year value is obtained using the UNIX system call TIME() which returns a long integer value for the current date and time. The C subroutine LOCALTIME() is used and returns a pointer to a structure containing the year in the form YY. The function then adds 1900 to the year value.

SUBSTR

Usage:

SUBSTR( s1, s2, start, len )

s1 = string returned
s2 = string to substring
start = position to start substring in s1
len = length of string required

Description:

This function returns a string of length LEN which is part of the string s1.
EDIT_CONV_INT

Usage:
EDIT_CONV_INT( s1, len, x )

s1 = string to convert
len = length of string s1
x = integer returned

Description:
This function converts a string of numbers s1 into its equivalent integer value. The string is first edited to make sure it contains only valid digits and is correctly terminated with a NULL (ie \0 ). If the string is valid the C function ATOI is used to convert the string to an integer.

ITOA

Usage:
ITOA( n, s1 )

n = integer to convert
s1 = converted string

Description:
This function converts an integer into a character string. Refer C Programming Manual for details.

GET_SELECTION

Usage:
code_return = GET_SELECTION( field, length )

field = the selection field.
length = length of field in characters.

CODE_RETURN - GET_SELECTION will return a code to indicate the type of control key used to terminate the input field. It may be set to the values
CD_RETURN, or
CD_TOP,
CD_BOTTOM,
CDCANCEL, these are initialised in the constants file - constants.h. This field may then be checked by the calling function to see how it should proceed.

Description:
Operates similarly to GETIELD but accepts a numeric selection for a subject in the ASSISTED_SEARCH procedure of L0401.c. It terminates the field and may set two extra values in CODE_RETURN if the up arrow (CD_TOP ) or down arrow (CD_BOTTOM ) keys are pressed.
GET_SUBJECT

Usage:

\[
\text{code\_return} = \text{GET\_SUBJECT}(\text{field}, \text{length})
\]

\[
\text{field} = \text{current screen field or data item.}
\]
\[
\text{length} = \text{length of field in characters.}
\]

CODE\_RETURN - GET\_SUBJECT will return a code to indicate the type of control key used to terminate the input field. It may be set to the values CD\_RETURN, or CD\_EXIT, CD\_CANCEL, these are initialised in the constants file - constants.h. This field may then be checked by the calling function to see how it should proceed.

Description:

Operates similarly to GETFIELD but returns the code CD\_EXIT when CNTL-E is pressed instead of returning to the calling process. This function is used in the L0203.c the Maintain Subjects function. The function rewrites the Subject Reference File before exiting.
CHECK_PARENT

Usage:

CHECK_PARENT (parent1, parent2, child1, child2)

parent1 = position of the subject to be checked currently
parent2 = position of the original subject to be checked as parent
child1 = position of the first child of a subject
child2 = position of the subject to be checked as child

FLAG_PARENT must be declared globally by the function calling CHECK_PARENT and initialised to zero (0) prior to each call. It will then be set to one (1) by CHECK_PARENT if parentage is detected and so may be checked on return to the calling function.

On the initial call to CHECK_PARENT then parent1 and parent2 should both be set to the subject we wish to see is an ancestor. Child1 and child2 should be the subject we wish to check as the descendant.

Description:

This function verifies if two subjects have a PARENT-CHILD relationship.

The algorithm would be as follows:-

If the subject to be checked currently equals the original child to be checked then

  Found to be descendant set FLAG_PARENT

If the subject to be checked as parent has children then

  call CHECK_PARENT with the child pointer of this subject
  the original subject remains the same
  the child pointer of this subject
  the original child to be checked remains the same

If the subject to be checked as parent is not the original subject and the subject to be checked as parent brother pointer does not equal the position of the first child then

  call CHECK_PARENT with the brother pointer of this subject
  the original subject remains the same
  the child pointer remains the same
  the original child to be checked remains the same
VAL_SUBJ_TAB

Usage:

\[
\text{flag-valid} = \text{VAL_SUBJ_TAB} (\text{subject, position, level})
\]

- \text{subject} = \text{the input subject field to be checked}
- \text{position} = \text{if the subject found returned as its position in the table else zero (0)}
- \text{level} = \text{if the subject found the level number of the subject in the tree as stored on the table else zero (0)}

\text{flag_valid} = \text{VAL_SUBJ_TAB} \text{ returns a flag, zero (0) if the subject was not in the tree and one (1) if it was.}

Description:

This function checks whether the passed subject is found in the Subject Reference table. It simply loops through the table elements checking if they equal the one passed. If a match is found then one (1) is returned to the calling function and the position and level numbers set, else zero is returned and position and level set to zero.
All programs contain a conditional test for flag-debug, as defined in the system constants file. When the flag is switched on messages are written to standard error. These messages provide a trace of the execution of programs and assist in locating bugs.

When any changes are made to a module in the system a file is provided (MAKEFILE) which recompiles and relinks all L.I.E.S modules. To recompile after changes are made type

MAKE LIES

All constants for the system have been placed in a single file constants.h which is included in the top of all programs. To change the value of any constant field this file need only be updated and the MAKEFILE executed to incorporate the change into all programs.

The length of any field may simply be varied in the above fashion, however care must be taken to ensure it does not upset the format of any screens or reports which require it. The length of the Subject field has the added restriction placed on it by the ASSISTED_SEARCH procedure, which is stated in section (5.1) of this report.

It should be noted that the length of all string fields have been declared as one longer than the length specified in the constants file. This is due to a requirement of the C Programming Language that string fields have a null character at the end. For ease of string manipulation we simply declare the string fields one character longer than required to cater for this null character.
10. FUTURE ENHANCEMENTS

The present implementation does not perform the following functions:

* Fines processing
* Reservation processing
* Closed Reserve processing
* Backup and Restore capabilities.

However, interfaces to these functions have been catered for in the design of LIES, and the current file design will not need to be changed to implement these future enhancements. For more information refer to the Systems Analysis document.

It would also be possible to make use of a barcode reader in the BORROW AN ITEM and RETURN AN ITEM functions for the inputting of borrower numbers and item and copy number.
### FILE DEFINITION OF BORROWER DATA FILE

```c
#include <stdio.h>

// Structure definition
typedef struct {  // Borrower's identity
    char identity[20];  // Borrower's unique identifier
} identity_borrower;

typedef struct {  // Borrower's data
    char surname[20];  // Borrower's surname
    char given_name[20];  // Borrower's given name
} data_borrower;

typedef struct {  // Borrower's contact information
    char number[20];  // Borrower's contact number
    char street[40];  // Borrower's street address
    char town[20];  // Borrower's town
    char postcode[20];  // Borrower's postcode
} contact_borrower;

typedef struct {  // Borrower's assets
    int item_number;  // Unique number for each item
    int item_copy_number;  // Number of copies for each item
    int item_amount;  // Amount for each item
} item_borrower;

typedef struct {  // Borrower's flag
    int flag_fines;  // Flags for fines
    int flag_discounts;  // Flags for discounts
    int flag_interest;  // Flags for interest
} flag_borrower;
```

### Description

1. **identity_borrower**: Contains the unique identifier of the borrower. It is limited to 20 characters.
2. **data_borrower**: Stores the surname and given name of the borrower. Each name is limited to 20 characters.
3. **contact_borrower**: Includes the contact number, street address, town, and postcode of the borrower. Each detail is limited to 20 characters.
4. **item_borrower**: Represents the item information, including item number, copy number, and amount. Each item is identified by a unique number and has a copy number for each item. The amount is also stored.
5. **flag_borrower**: Flags for fines, discounts, and interest, each limited to 1 bit (0 or 1) for each flag.
FILE DEFINITION OF ON LOAN DATA FILE

struct
{
    int item_number;
    int copy_number;
    int identity_borrower;
    int date_due_back;
};
FILE DEFINITION OF CATALOGUE TRANSACTION FILE

struct trd
{
    char transaction_code ;
    long date_time_transaction ;
    int item_number ;
    char medium ;
    struct surname LTH_SURNAME +11 ;
    ( char given_name LTH_GIVEN_NAME +11 ;
    ) data_given_names [ MAX_GIVEN_NAMES ] ;
    ) data_authors MAX_AUTHORS ] ;
    char title_name LTH_TITLE +11 ;
    int date_published ;
    int edition_number ;
    char publisher_name LTH_PUBLISHER +11 ;
    char dewey_number LTH_DEWEY +11 ;
    struct subject_identifier LTH_SUBJECT +11 ;
    ) data_subjects MAX_SUBJECTS ] ;
    int number_copies ;
} ;
FILE DEFINITION OF SUBJECT TRANSACTION FILE

struct trans_subj
{
    char transaction_code[ LTH_TRAN_CD +1 ] ;
    char old_subject [ LTH_SUBJECT +1 ] ;
    char new_subject [ LTH_SUBJECTC +1 ] ;
    int offset_subj_index ;
} ;
FILE DEFINITION OF BORROWER TRANSACTION FILE

struct borrower {
  char surname[11];
  char given_name[9];
  char street_name[14];
  char suffix_name[9];
  char phone[11];
  char LTH_NAME[11];
  char LTH_SURNAME[11];
  char LTH_GIVEN_NAME[9];
  char LTH_POSTCODE[11];
};

#define END 11
#define LTH_NAME 11
#define LTH_SURNAME 11
#define LTH_GIVEN_NAME 9
#define LTH_POSTCODE 11

/* Additional comments or documentation */
struct out, struct
char *lname, char *name, char *given, given
Surname LTH GIVEN NAME

FILE DEFINITION OF AUTHOR INDEX FILE

long offset, catalogue

medium, publication

FILE DEFINITION OF TITLE INDEX FILE

struct title_ix
    {char
title_name[LTITLE + 1];
    char
title
    char
medium;
    int
date_published;
    long
offset_catalogue;
}
FILE DEFINITION OF RESULTS FILE

struct

int anumber; // item number
char title[LTH_TITLE + 1];
char author[LTH_AUTHOR + 1];
char place[LTH_PLACE + 1];
char date[LTH_DATE + 1];
char givenname[LTH_GIVEN_NAME + 17];
char surname[LTH_SURNAME + 17];
char full_name[LTH_GIVEN_NAME + LTH_SURNAME + 17];

int total; // total number of items
int numAuthors; // number of authors
int numPublisher; // number of publishers
int numSuburb; // number of suburbs
int numPostcode; // number of postcodes

/* DO NOT MODIFY THIS FILE */
# 2. CONSTANTS

/* constants */

/* GENERAL */

#define flag_debug 0  // Turn on/off debug traces */
#define ACCEPT 1  // Acceptable screen character */
#define REJECT 9  // Reject the screen character */

#define NULL_STRING ""  // Null string */
#define BLANK_STRING ""  // Blank string */

#define NULL_CHARACTER "\0"  // Null character */
#define BLANK_CHARACTER "\t"  // Blank character */

#define NULL_OFFSET -1  // Null offset in Subject table */
#define NULL_PTR -1  // Null pointer in Subject table */

#define QUESTION_MARK "?"  // Character question mark */

#define MAX_SEARCH_SUBJECTS 3  // Max subjects to search simultaneously */
#define MAX_DAYS_ON_1_DAY 604800  // 7 days represented in seconds */

/* SYSTEM ERROR CONDITIONS */

#define STATUS_READ_ERROR 10
#define STATUS_WRITE_ERROR 20
#define STATUS_OPEN_ERROR 30
#define STATUS_CLOSE_ERROR 40
#define STATUS_SORT_ERROR 50
#define STATUS_CREATE_ERROR 60
#define STATUS_HD_SUBJECT 70
#define STATUS_SEARCH_ERROR 80

/* STACK CONSTANT */

#define STACK_SIZE 5

/* POSITION OF THE ROOT SUBJECT IN THE SUBJECT TABLE */

#define ROOT_PTR 0
HEART CURE

+ Return an item from storage
+ Drop an item from storage

TRANSACTION CODES

+ Print CP TOP
+ Set transaction CP
+ Get transaction CP
+ Enter transaction CP
+ Return transaction CP
+ Display transaction CP

SCREEN CONTROL KEYPAD

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
<td>Tab Key</td>
</tr>
<tr>
<td>!</td>
<td>Tab Key</td>
</tr>
<tr>
<td>#</td>
<td>Tab Key</td>
</tr>
<tr>
<td>$</td>
<td>Tab Key</td>
</tr>
<tr>
<td>%</td>
<td>Tab Key</td>
</tr>
<tr>
<td>^</td>
<td>Tab Key</td>
</tr>
<tr>
<td>&amp;</td>
<td>Tab Key</td>
</tr>
<tr>
<td>'</td>
<td>Tab Key</td>
</tr>
<tr>
<td>(</td>
<td>Tab Key</td>
</tr>
<tr>
<td>)</td>
<td>Tab Key</td>
</tr>
<tr>
<td>-</td>
<td>Tab Key</td>
</tr>
<tr>
<td>_</td>
<td>Tab Key</td>
</tr>
<tr>
<td>+</td>
<td>Tab Key</td>
</tr>
<tr>
<td>=</td>
<td>Tab Key</td>
</tr>
<tr>
<td>]</td>
<td>Tab Key</td>
</tr>
<tr>
<td>[</td>
<td>Tab Key</td>
</tr>
<tr>
<td>\</td>
<td>Tab Key</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>Tab Key</td>
</tr>
<tr>
<td>?</td>
<td>Tab Key</td>
</tr>
<tr>
<td>`</td>
<td>Tab Key</td>
</tr>
<tr>
<td>a</td>
<td>Tab Key</td>
</tr>
<tr>
<td>b</td>
<td>Tab Key</td>
</tr>
<tr>
<td>c</td>
<td>Tab Key</td>
</tr>
<tr>
<td>d</td>
<td>Tab Key</td>
</tr>
<tr>
<td>e</td>
<td>Tab Key</td>
</tr>
<tr>
<td>f</td>
<td>Tab Key</td>
</tr>
<tr>
<td>g</td>
<td>Tab Key</td>
</tr>
<tr>
<td>h</td>
<td>Tab Key</td>
</tr>
<tr>
<td>i</td>
<td>Tab Key</td>
</tr>
<tr>
<td>j</td>
<td>Tab Key</td>
</tr>
<tr>
<td>k</td>
<td>Tab Key</td>
</tr>
<tr>
<td>l</td>
<td>Tab Key</td>
</tr>
<tr>
<td>m</td>
<td>Tab Key</td>
</tr>
<tr>
<td>n</td>
<td>Tab Key</td>
</tr>
<tr>
<td>o</td>
<td>Tab Key</td>
</tr>
<tr>
<td>p</td>
<td>Tab Key</td>
</tr>
<tr>
<td>q</td>
<td>Tab Key</td>
</tr>
<tr>
<td>r</td>
<td>Tab Key</td>
</tr>
<tr>
<td>s</td>
<td>Tab Key</td>
</tr>
<tr>
<td>t</td>
<td>Tab Key</td>
</tr>
<tr>
<td>u</td>
<td>Tab Key</td>
</tr>
<tr>
<td>v</td>
<td>Tab Key</td>
</tr>
<tr>
<td>w</td>
<td>Tab Key</td>
</tr>
<tr>
<td>x</td>
<td>Tab Key</td>
</tr>
<tr>
<td>y</td>
<td>Tab Key</td>
</tr>
<tr>
<td>z</td>
<td>Tab Key</td>
</tr>
<tr>
<td>{</td>
<td>Tab Key</td>
</tr>
<tr>
<td>}</td>
<td>Tab Key</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>Tab Key</td>
</tr>
<tr>
<td>?</td>
<td>Tab Key</td>
</tr>
<tr>
<td>`</td>
<td>Tab Key</td>
</tr>
</tbody>
</table>

SCREEN KEY DEFINITIONS

(© 1987 (})
HELP SCREEN IDENTITIES

#define S01 1
#define S02 2
#define S0301 201
#define S0202 202
#define S0203 203
#define S0204 204
#define S0205 205
#define S0206 206
#define S03 3
#define S0301 301
#define S0302 302
#define S0303 303
#define S030301 30301
#define S0305 305
#define S0401 401
#define S0402 402
#define S0403 403

LENGTH OF SCREEN FIELDS

#define LTH_COPIES 3
#define LTH_DATE 4
#define LTH_DEWEY 11
#define LTH_EDITION 2
#define LTH_ERR_MSG 70
#define LTH_GIVEN_NAME 10
#define LTH_ITEM_NO 6
#define LTH_MEDIUN 1
#define LTH_MUN_ST 7
#define LTH_PASSWORD 8
#define LTH_POSTCODE 4
#define LTH_PUBLISHER 96
#define LTH_SELECTION 2
#define LTH_ST_NAME 30
#define LTH_SUBJECT 15
#define LTH_SUB_NAME 26
#define LTH_SUBJ_DESC 35
#define LTH_SURNAME 20
#define LTH_TEL_NUM 11
#define LTH_TEXT_MSG 70
#define LTH_TTLE 60
#define LTH_TODAY 12
#define LTH_TRNS_CD 1
#define LTH_BORR_NO 7
### 11.3 STANDARD SCREEN LAYOUT

Each screen in the L.I.E.S has the standard layout pictured above.

<table>
<thead>
<tr>
<th>Field</th>
<th>Name</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Current-date</td>
<td>DD/MM/YY</td>
<td>The current date</td>
</tr>
<tr>
<td>B</td>
<td>Current-time</td>
<td>HH:SS</td>
<td>The current time</td>
</tr>
<tr>
<td>C</td>
<td>Error-message</td>
<td>Char(80)</td>
<td>Error message generated by the process</td>
</tr>
<tr>
<td>D</td>
<td>Text-message</td>
<td>Char(80)</td>
<td>Text message generated by the process</td>
</tr>
</tbody>
</table>

Line 23 of each screen contains the control key definitions for that function. Two standard definitions for most screens are:

- **CTL-A** - to display help information for a function
- **CTL-E** - to return to the previous menu screen
11.4 FILE NAMES

PROGRAMS

L01.c  L.I.E.S logo
L0101.c Create Subject Table
L0102.c Manager Access Check
L0103.c Librarian Access Check

L02.c  Manager Functions Menu
L0201.c Maintain Catalogue
L0202.c Maintain Borrower
L0203.c Maintain Subjects
L0204.c Submit File Changes Menu
L0204010101.c Sort Catalogue & Remove Redundant Transactions
L0204010103.c Perform Catalogue Update
L0204010105.c Update Subjects on Catalogue
L02040102.c Create Catalogue Indexes
L02040103a.c Sort Author Index
L02040103b.c Sort Title Index
L02040103c.c Sort Subject Index
L02040104.c Recreate Subject Reference File
L0204020101.c Sort Borrower & Remove Redundant Transactions
L0204020103.c Perform Borrower Update
L02040202.c Create Borrower Indexes
L0204030101.c Sort Daily Trans & Remove Redundant Transactions
L0204030103.c Perform Daily Update
L0205.c Management Reports Menu
L020501.c Catalogue Report
L020502.c Borrower Report
L020503.c Onloan Report
L020504.c Overdue Report
L0206.c Change Manager Password

L03.c  Librarian Functions Menu
L0301.c Borrow an Item
L0302.c Return an Item
L0303.c Display Borrower Information
L030301.c Collect Search Criteria
L030302.c Process Borrower Search
L030303.c Display Borrower Search Results
L0304.c Print Subject Definitions
L0305.c Change Librarian Password

L04.c  Catalogue Data Retrieval
L0401.c Collect Search Criteria
L0402.c Process Catalogue Search
L0403.c Display Catalogue Search Results
L040301.c Print Catalogue Search Results
HELP FILES

S01.hlp Logo
S02.hlp Manager Functions
S0201.hlp Maintain Catalogue
S0202.hlp Maintain Borrower
S0203.hlp Maintain Subjects
S0204.hlp Submit File Changes
S0205.hlp Management Reports
S0206.hlp Change Manager Password
S03.hlp Librarian Functions
S0301.hlp Borrow an Item
S0302.hlp Return an Item
S0303.hlp Display Borrower Information
S0304.hlp Print Subject Definitions
S0305.hlp Change Librarian Password
S0401.hlp Catalogue Data Retrieval
S0402.hlp Assisted Search
S0403.hlp Display Results

FILE STRUCTURES

catalogue.h Catalogue File
borrower.h Borrower File
onloan.h On Loan File
author_index.h Author Index File
title_index.h Title Index File
subj_index.h Subject Index File
bor_index.h Borrower Index File
trans_bor.h Borrower Transaction File
trans_cat.h Catalogue Transaction File
trans_daily.h Daily Transaction File
trans_subj.h Subject Transaction File
results.h Search Results File
subj_ref.h Subject Reference File

DATA FILES

catalogue.dat Catalogue File
borrower.dat Borrower File
onloan.dat On Loan File
auth_index.dat Author Index File
tit_index.dat Title Index File
sub_index.dat Subject Index File
bor_index.dat Borrower Index File
trans_bor.dat Borrower Transaction File
cat_tran.dat Catalogue Transaction File
daily_tran.dat Daily Transaction File
subj_tran.dat Subject Transaction File
results.dat Search Results File
subj_ref.dat Subject Reference File

SHELL FILES

L020401.sh Catalogue Update Sequence
L020402.sh Borrower Update Sequence
L020403.sh On-Loan Update Sequence
L020404.sh Update Sequence for all Files
GLOBAL FUNCTIONS

CALCDATE.c  Calculate Current Date
CURRENT_YEAR.c  Calculate Current Year
EDITCONVINT.c  Edit and Convert String into Integer
GETFIELD.c  Get a Field From the Screen
GETPWD.c  Get Password From Screen
GETSEL.c  Get Numeric Selection From Screen
GET_SUBJECT.c  Get Subject From Screen
HELPINFO.c  Display Help Information
ITOA.c  Converts Integer to String
SUBSTR.c  Substring of a String
CHECK_PARENT.c  Check if one passed subject is ancestor of the other in the Subject tree
VAL_SUBJ_TAB.c  Check if the passed subject occurs in the Subject tree

BINARY SEARCHES

BINCAT.c  Binary search of Catalogue File
BINAUTH.c  Binary search of Catalogue Author Index
BINTITLE.c  Binary search of Catalogue Title Index
BINBORR.c  Binary search of Borrower File
BINNAME.c  Binary search of Borrower Name Index
BINONLOAN.c  Binary search of On Loan File
NAME
LIES - library information and enquiry system

SYNOPSIS
LIES

DESCRIPTION
LIES is a Library Information and Enquiry System that provides a user friendly interface and fast access to both the catalogue and borrower information stored in the system.

The user friendly interfaces are provided through the use of tailored menus for the different users of the system:

* the borrowers,
* the librarian,
* the library manager,

and screen based online help for all functions.

FILES
/u/p/321/g1/LIES

AUTHORS
Kerrie Barker
Helen Bow
Andrew Byrnes
Vicki Grossmann
Allan Hird

LIMITATIONS
Due to internal sort routines, transaction files are limited to 10,000 records and index files are limited to 40,000. These limits are a consequence of the size of the library specified for the system and can easily be increased. To increase these limits see LIES system constants file.

BUGS
Multiple users updating files simultaneously may create file inconsistencies.
| LLLL | I I I I I I I I | E E E E E E E E | S S S S S S S S S S |
| LLLL | I I I I I I I I | E E E E E E E E | S S S S S S S S S S |
| LLLL | I I I | E E E | S S S S S S |
| LLLL | I I I | E E E E E E E E | S S S S S S S S S S |
| LLLL | I I I | E E E E E E E E | S S S S S S S S S S |
| LLLL | I I I | E E E | S S S S S S |
| LLLLLLLLLL | I I I I I I I | E E E E E E E E | S S S S S S S S S S |
| LLLLLLLLLL | I I I I I I I | E E E E E E E E | S S S S S S S S S S |

( LIBRARY INFORMATION & ENQUIRY SYSTEM )
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>1. CATALOGUE INFORMATION RETRIEVAL</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Assisted Search</td>
<td>5</td>
</tr>
<tr>
<td><strong>2. LIBRARIAN FUNCTIONS</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Borrow an Item</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Return an Item</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Display Borrower Information</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Print Subject Definitions</td>
<td>8</td>
</tr>
<tr>
<td>2.5 Change Password</td>
<td>8</td>
</tr>
<tr>
<td><strong>3. LIBRARY MANAGER FUNCTIONS</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Maintain Catalogue</td>
<td>10</td>
</tr>
<tr>
<td>3.2 Maintain Borrower</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Maintain Subjects</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Submit File Changes</td>
<td>13</td>
</tr>
<tr>
<td>3.5 Management Reports</td>
<td>14</td>
</tr>
<tr>
<td>3.6 Change Password</td>
<td>14</td>
</tr>
</tbody>
</table>
INTRODUCTION

L.I.E.S. (Library Information and Enquiry System) is a system that has been divided into three broad functional areas:

- Borrower
- Librarian
- Library Manager

These three areas correspond to the three users for which the system was designed. It is assumed that the Borrower should only be allowed to perform catalogue information retrievals; the Librarian should be allowed to perform the Librarian functions (plus the Borrower functions); and the Library Manager should be allowed to perform the Library Manager functions (plus the Borrower functions, and, if need be, the Librarian functions).

For the above reasons the User Manual is similarly broken up into three detachable booklets, such that the relevant sections may be distributed to the users indicated.

Because of the broad spectrum of users L.I.E.S. is designed to accommodate, user friendliness is essential. For this reason such features as menus, help screens, User Manuals and tutorial Manuals have been used to assist the novice. On the other hand, there are many advanced features of L.I.E.S. that will also satisfy the needs of the experienced user.

As just stated L.I.E.S. is a menu-driven system. The primary menu is the L.I.E.S. logo. From this logo any of the functions of L.I.E.S. may be accessed (with the correct security).

L.I.E.S. resides entirely on the UNIX supported Pyramid machine and may be quite simply invoked by typing "LIE" and pressing <RETURN>. When the L.I.E.S. logo screen is returned the user may invoke the Catalogue Information Retrieval function by pressing <RETURN>. To invoke the Librarian or Library Manager functions the user must be aware of certain security practices, and so should consult the Library Manager if unsure of a logon procedure.

If the user presses any other key then a help screen will be returned to instruct the user on how to enter L.I.E.S.
Once the user has entered L.I.E.S. there are a number of screen handling features that should firstly be understood before you can gain maximum efficiency from the L.I.E.S. system. A brief description of most of these features is given below but if further information is needed it is suggested that you consult the Tutorial Manual.

To move around the screen use the <TAB> key to move from field to field and the left and right arrow keys for movements within a field. When the cursor is placed on the field where you need to enter data you may begin typing. If you make a mistake there are a number of ways in which you can correct the error. Firstly, you can move the cursor back to the error (by using the left or right arrows) and then simply overtyping the error. Other alternatives could include a combination of tabbing and other features listed below.

To erase a character the user should use the delete key and if he/she wishes to erase the whole field he/she can use the <ESC> (erase screen component) key.

To position the cursor at the beginning of a field the user should press the <HOME> key.

To enter a function from a menu the user should simply type in the appropriate number for that function. (NOTE: there is no need to press <RETURN> after entering your selection). The function will then be returned ready for your use. To exit from the function to the menu screen the user should press <CNTL-E> (EXIT). (To use a CNTL-key hold down the <CNTL> key and press the character you require).

At any time if the user requires help on how to use the function they see in front of them they may call up a brief tutorial (help screen) by pressing <CNTL-A> (ASSIST).
1. CATALOGUE INFORMATION RETRIEVAL

The Catalogue Information Retrieval function enables the user to enter and effect a search on Library catalogue items. Catalogue items may include any source of information that is kept in the Library no matter what the form. (i.e. could include books, software, magazines, journals, tapes, etc.).

The function is not security protected, as it is envisaged that anyone might want to use this function to retrieve information on items in the Library.

To access the Catalogue Information Retrieval function simply hit <RETURN> or <ENTER> on the L.I.E.S. logo screen. This will then return the Catalogue Information Retrieval screen ready for your use.

The user can perform a search on items within the Library by specifying what sort of information he/she so requires in the given fields. A search may be performed on any of the following criteria (or a conjunction thereof):

- **Author**: the author surname and optionally the given name(s) or initial(s).

- **Title**: the exact title name.

- **Medium**: the type of item (i.e. book, tape, journal, etc).

- **Date Published**: greater than (or equal to) a date published, less than (or equal to) a date published, or a range of dates.

- **Subject**: a list of up to three (3) subjects that will be used conjunctionally.

Note: Medium and Date Published must be in conjunction with one of the other search criteria.

Searches on a particular Author and/or Title (which may be restricted by Medium and/or Date Published) are straightforward in that you have asked for something quite specific. Such a search is common place to the regular library user because this information may have been available (although not nearly as quick) by using the old cards system. L.I.E.S. however has an added feature that will allow the user to retrieve information on a specified subject. It is this feature that will be explained in greater detail because it is something that needs more understanding in order to operate it efficiently.
The subject search enables the user to ask for all material on a certain subject, however, the user must know what subjects are allowable. L.I.E.S. contains a subject tree, whereby, a predetermined list of subjects is set up for the user to choose from. To view this subject tree the user has two options:

i) ask the Librarian for a copy of the subject definitions.

ii) use the Assisted search facility (for more information see section 1.1 Assisted Search).

The user has the option of getting a printout of the retrieved items. To do this the user must press <CNTL-P> on the first page of items retrieved on the screen. For more detail on the layout of the report see Figure R-01 (Report Example) given after the screen examples following.
1.1 ASSISTED SEARCH

The Assisted Search facility guides the user through the allowable subjects until he/she finds the subject that will best suit their needs. These allowable subjects are kept in what is referred to as a 'Subject Tree'. This subject tree is set up such that the higher levels contain the broad subject categories and the lower down the tree you travel the more specific the subjects become. For example the first level of the tree may contain the subject 'DATA'. Now the user may view what subjects 'DATA' covers by looking at its children. One of its children may be 'DATABASE' for which the user is interested. On viewing its children, 'DB ADMIN' and 'DB STORAGE', the user may then decide that 'Database' was specific enough and so may return back up the tree to select 'DATABASE' as the search subject.

To invoke the Assisted Search facility the user types in a '?' in the first character of the first subject field and presses <RETURN>. This will then return the first level of subjects of the subject tree. Upon viewing the subjects in the first level the user may want to see what subjects are contained below a given subject. To view the subjects below, the user types the number given beside the subject he/she wants to peruse in the Subject Selection field and presses the 'Down arrow'. The subjects on the next level will then be displayed for the user's perusal. If the user wants to move down the tree further he/she uses the same procedure as just described with the 'Down arrow'. When the bottom of the tree is reached for a subject area a suitable message will be displayed indicating that no further subjects exist. If the user wants to go back up the tree he/she simply presses the 'Up arrow' to return to the previous level.

When the user has finally found the subject that he/she wishes to search on there are two ways in which they can invoke the search of that particular subject:

i) Type in the number corresponding to the subject he/she wishes to search on in the Subject Selection field and press <RETURN>.

ii) If the user wishes to restrict the search by medium, for example, he/she must press <CNTL-C> (CANCEL), to cancel the Assisted Search facility and then manually input the medium and the subject to be searched on before pressing <RETURN>. 
2. LIBRARIAN FUNCTIONS

To enter the Librarian menu type '2' on the L.I.E.S. logo screen. This will then return a screen prompting for the password needed to enter the Librarian functions. After typing in the password required hit <RETURN>; this will then return the Librarian menu ready for use.

The functions available to the librarian are:

1. Borrow an Item
2. Return an Item
3. Borrower Information Retrieval
4. Print Subject Definitions
5. Change Password

Each of the five functions seen in the Librarian menu may be selected by inputting the relevant number of the function required. (e.g. if the Librarian wishes to Return an Item he/she would simply type '2'). This will then return the required function ready for use.

To return to the L.I.E.S. logo screen press <CNTL-E> (EXIT).

2.1 Borrow an Item

This function allows the librarian to enter information about an item being borrowed. The librarian enters:
- Borrower number
- Catalogue number
- Copy number

After the information is entered three checks are performed:
- has the borrower any fines owing
- has the borrower 10 items on loan already
- is the item currently on loan

If all checks are passed then a message will be returned indicating that the item was borrowed successfully. Otherwise an error message will be returned indicating what check was not passed.

To return to the Librarian menu press <CNTL-E> (EXIT).
2.2 Return an Item

Introduction

This function allows the librarian to enter information about an item being returned. The librarian enters:

- Catalogue number
- Copy number

After the information is entered one check is performed:

- is the item currently on loan.

A message will be returned indicating that the item was returned successfully. Otherwise an appropriate error message will be returned.

To return to the Librarian menu press <CNTL-E> (EXIT).

2.3 Borrower Information Retrieval

This function will allow the Librarian to retrieve borrower data, including a list of items currently on loan for that borrower and the date they are due back. The Librarian must enter the borrower number or the borrower's surname (with or without the borrower's given name/s or initials).

Because the borrower number is a unique identity the function will return all relevant data for that borrower. However, a borrower surname may not be unique, and so if there is more than one borrower with that surname the Librarian must specify what borrower he/she wants information on. Hence, if there is more than one borrower with that surname the function will return them one at a time, listing their full given name/s and address. With this information the Librarian can select which borrower he/she wants information for. When the correct borrower is displayed the Librarian must then hit <RETURN>; with the borrower number now known the function will then return all the items that borrower has on-loan and the date they are due back.

If the Librarian performs a search on borrower name and there is only one borrower in the library with that name then information on what books that borrower has on-loan will be returned immediately (i.e. without the Librarian having to hit <RETURN> again).

If any item is overdue then the date due back (on the screen) will be highlighted for the Librarian.

The information returned to the screen on the items borrowed is a condensed version; to gain a more detailed list the Librarian may request a printed list of the items by pressing <CNTL-P> (PRINT) when the items are displayed on the screen.

To return to the Librarian menu press <CNTL-E> (EXIT).
2.4 Print Subject Definitions

To print the Subject definitions the Librarian should select '4' in the Librarian menu screen. This will then print the report on the local printer.

The report is formatted such that you can see what subjects 'come under' other subjects, (i.e. you can see the parent child relationships between the subjects).

From this report you can see that a first level subject might be 'DATA'. Underneath 'DATA' (i.e. 'DATA's children) are 'FILES' and 'DATABASE', for example, and underneath 'DATABASE' is 'DB DESIGN'. This is how the subject tree is structured.

2.5 Change Password

To change the Librarian password the Librarian MUST know the existing password.

To change the password the Librarian must type in both the existing password and the new password and press <RETURN>. After checking that the Librarian has entered the existing password correctly the function will then request that the Librarian re-type the new password to ensure that it was typed correctly in the first place. After re-typing the new password and pressing <RETURN> the password will be changed and a message returned indicating this.

To return to the Librarian menu press <CNTL-E> (EXIT).
3. LIBRARY MANAGER FUNCTIONS

To enter the Library Manager menu type '1' on the L.I.E.S. logo screen. This will then return a screen prompting for the password needed to enter the Library Manager functions. After typing in the password required hit <RETURN>; this will then return the Library Manager menu ready for use.

The functions available to the Library Manager are:

1. Maintain Catalogue
2. Maintain Borrower
3. Maintain Subjects
4. Submit File Changes
5. Management Reports
6. Change Password

Each of the six functions seen in the Library Manager menu may be selected by inputing the relevant number of the function required. (eg. if the Library Manager wishes to Maintain Catalogue Data he/she would simply type '1'). This will then return the required function ready for use.

To return to the L.I.E.S. logo screen press <CNTL-E> (EXIT).
3.1 Maintain Catalogue Data

This function enables the Library Manager to Maintain Catalogue Information and may be accessed from the Library Manager menu by selecting '1'.

When the Maintain Catalogue Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Catalogue Record.

To display a catalogue record type in the catalogue number to be displayed and press <RETURN>.

To create a catalogue record place a 'C' in the Transaction field, enter the relevant information for that catalogue record, and press <RETURN>. If the catalogue record does not already exist and all information has been entered correctly then a message will be displayed indicating that the record was created successfully; otherwise an appropriate error message will be displayed.

Before a catalogue record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'U' in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a catalogue record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'D' in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.

When a record is either created, updated or deleted the transaction is written to the Catalogue Transaction File for processing at a later date. The transactions may be written to the Catalogue file by Submitting File Changes (see Section 3.4 for more detail).

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTL-E> (EXIT).
3.2 Maintain Borrower Data

This function enables the Library Manager to Maintain Borrower Information and may be accessed from the Library Manager menu by selecting '2'.

When the Maintain Borrower Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Borrower Record.

To display a borrower record type the borrower number to be displayed and press <RETURN>.

To create a borrower record place a 'C' in the Transaction field, enter the relevant information for that borrower record, and press <RETURN>. If the borrower record does not already exist and all information has been entered correctly then a message will be displayed indicating that the record was created successfully; otherwise an appropriate error message will be displayed.

Before a borrower record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'U' in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a borrower record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'D' in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.

When a record is either created, updated or deleted the transaction is written to the Borrower Transaction File for processing at a later date. The transactions may be written to the Borrower file by Submitting File Changes (see Section 3.4 for more detail).

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTL-E> (EXIT).
3.3 Maintain Subject Reference Data

The Subject Reference File is kept in what is referred to as a 'Subject Tree'. This subject tree is set up such that the higher levels contain the broad subject categories and the lower down the tree you travel the more specific the subjects become. For example the first level of the tree may contain the subject 'DATA'. Now below 'DATA' may be the subjects 'DATA STRUCTURES', 'DATA CODING', 'FILES', 'DATABASE' and 'DATA STORAGE'. Then below 'DATABASE' may be the subjects 'DB DESIGN', 'DB MANAGEMENT', 'DB SYSTEMS' and 'DB ADMIN'. In this example the subject 'DATA' is the parent of the subject 'DATABASE', and 'DATABASE' is the parent of the subject 'DB ADMIN'. This also implies that the subject 'DB ADMIN' is a child of 'DATABASE', and 'DATABASE' is a child of the subject 'DATA'.

The 'root' of the tree is the highest level of the tree and will not have a parent subject of course; but this subject has been previously set up in creating the tree. We assume that the subject 'COMPUTING' will always be the 'root' of the tree and so any subject added will have a parent.

This function enables the Library Manager to Maintain Subject Information and may be accessed from the Library Manager menu by selecting '3'.

When the Maintain Subject Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Subject Record.

To display a subject record type in the Subject to be displayed and press <RETURN>.

To create a subject record place a 'C' in the Transaction field, enter the new subject, the parent of that subject, the description of the new subject and press <RETURN>. If the subject does not already exist and the parent subject does exist then a message will be displayed indicating that the subject was created successfully; otherwise an appropriate error message will be displayed. (If the subject being created is to be a first level subject then its parent is 'computing').

Before a subject record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'U' in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a subject record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'D' in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.
When a subject is either created, updated or deleted the subject reference file is updated immediately.

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTL-E> (EXIT).

3.4 Submit File Changes

This menu type function enables the Library Manager to submit file changes to update the Catalogue, Borrower or On-Loan Transaction Files.

The user has the option to request that one file be updated or he/she may request that all files be updated.

The On-Loan transaction file contains the days borrowing and returns transactions. This will be run upon request, which is assumed will be a nightly task.

The Borrower transaction file contains the transactions for new, updated or deleted borrowers since the last time the Borrower changes were submitted. This will be run upon request, which is assumed will be on the night following a new borrower being added.

The Catalogue transaction file contains the transactions for new, updated or deleted catalogue records since the last time the Catalogue changes were submitted. This will be run upon request, which is assumed will be once every six months or by special request.

To return to the Library Manager menu press <CNTL-E> (EXIT).
3.5 Management Report Generator

This menu type function enables the user to select any one of the management reports offered by the system. These reports are:

1. CATALOGUE ENTRY REPORT - This is a report of all the items currently on the CATALOGUE file.

2. BORROWER REPORT - This is a report of all the borrowers that are currently members of the library.

3. CURRENT ON LOAN REPORT - This is a report of all the items currently on loan.

4. CURRENT OVERDUE REPORT - This is a report of all items that are overdue.

To select the report required the user must input the appropriate selection (i.e. 1, 2, 3, or 4). This will then submit the report and print it at the local printer.

If there is no data found for the particular report, an empty report will be written to notify the user of this.

To return to the Library Manager menu press <CNTL-E> (EXIT).

1.6 Change Password

To change the Library Manager password the Library Manager MUST know the existing password.

To change the password the Library Manager must type in both the existing password and the new password and press <RETURN>. After checking that the Library Manager has entered the existing password correctly the function will then request that the Library Manager re-type the new password to ensure that it was typed correctly in the first place. After re-typing the new password and pressing <RETURN> the password will be changed and a message returned indicating this.

To return to the Library Manager menu press <CNTL-E> (EXIT).
( LIBRARY INFORMATION & ENQUIRY SYSTEM )

***************
* TUTORIAL *
* MANUAL *
***************
# TABLE OF CONTENTS

## INTRODUCTION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. CATALOGUE INFORMATION RETRIEVAL</td>
<td>1.1</td>
</tr>
<tr>
<td>1.1 Assisted Search</td>
<td>1.1.1</td>
</tr>
<tr>
<td>2. LIBRARIAN FUNCTIONS</td>
<td>2.1</td>
</tr>
<tr>
<td>2.1 Borrow an Item</td>
<td>2.1.1</td>
</tr>
<tr>
<td>2.2 Return an Item</td>
<td>2.2.1</td>
</tr>
<tr>
<td>2.3 Display Borrower Information</td>
<td>2.3.1</td>
</tr>
<tr>
<td>2.4 Print Subject Definitions</td>
<td>2.4.1</td>
</tr>
<tr>
<td>2.5 Change Password</td>
<td>2.5.1</td>
</tr>
<tr>
<td>3. LIBRARY MANAGER FUNCTIONS</td>
<td>3.1</td>
</tr>
<tr>
<td>3.1 Maintain Catalogue</td>
<td>3.1.1</td>
</tr>
<tr>
<td>3.2 Maintain Borrower</td>
<td>3.2.1</td>
</tr>
<tr>
<td>3.3 Maintain Subjects</td>
<td>3.3.1</td>
</tr>
<tr>
<td>3.4 Submit File Changes</td>
<td>3.4.1</td>
</tr>
<tr>
<td>3.5 Management Reports</td>
<td>3.5.1</td>
</tr>
<tr>
<td>3.6 Change Password</td>
<td>3.6.1</td>
</tr>
</tbody>
</table>
INTRODUCTION

L.I.E.S. (Library Information and Enquiry System) is a system that has been divided into three broad functional areas:

- Borrower
- Librarian
- Library Manager

These three areas correspond to the three users for which the system was designed. It is assumed that the Borrower should only be allowed to perform catalogue information retrievals; the Librarian should be allowed to perform the Librarian functions (plus the Borrower functions); and the Library Manager should be allowed to perform the Library Manager functions (plus the Borrower functions, and, if need be, the Librarian functions).

For the above reasons the User Manual is similarly broken up into three detachable booklets, such that the relevant sections may be distributed to the users indicated.

Because of the broad spectrum of users L.I.E.S. is designed to accommodate, user friendliness is essential. For this reason such features as menus, help screens and User Manuals have been used to assist the novice. On the other hand, there are many advanced features of L.I.E.S. that will also satisfy the needs of the experienced user.

As just stated L.I.E.S. is a menu-driven system. The primary menu is what is seen as the L.I.E.S. logo. From this logo any of the functions of L.I.E.S. may be accessed (with the correct security). An example of the L.I.E.S. primary menu logo screen can be seen in Figure S-01 overleaf.

L.I.E.S. resides entirely on the UNIX supported Pyramid machine and may be quite simply invoked by typing "LIES" and pressing <RETURN>. When the L.I.E.S. logo screen is returned the user may invoke the Catalogue Information Retrieval function by pressing <RETURN>. To invoke the Librarian or Library Manager functions the user must be aware of certain security practises, and so should consult the Library Manager if unsure of a logon procedure.

If the user presses any other key then a help screen will be returned to instruct the user on how to enter L.I.E.S.
This is the LIES logo screen (the primary menu screen).

To use a CNTL key hold the <CNTL> key down and press the letter that you require.
1. CATALOGUE INFORMATION RETRIEVAL

The Catalogue Information Retrieval function enables the user to enter and effect a search on Library catalogue items. Catalogue items may include any source of information that is kept in the Library no matter what the form. (i.e. could include books, software, magazines, journals, tapes, etc.).

The function is not security protected, as it is envisaged that anyone might want to use this function to retrieve information on items in the Library.

To access the Catalogue Information Retrieval function simply hit <RETURN> or <ENTER> on the L.I.E.S. logo screen. This will then return the Catalogue Information Retrieval screen ready for your use.

The user can perform a search on items within the Library by specifying what sort of information he/she so requires in the given fields. A search may be performed on any of the following criteria (or a conjunction thereof):

- **Author**: the author surname and optionally the given name(s) or initial(s).

- **Title**: the exact title name.

- **Medium**: the type of item (i.e. book, tape, journal, etc).

- **Date Published**: greater than (or equal to) a date published, less than (or equal to) a date published, or a range of dates.

- **Subject**: a list of up to three (3) subjects that will be used conjunctionally.

Note: Medium and Date Published must be in conjunction with one of the other search criteria.

Searches on a particular Author and/or Title (which may be restricted by Medium and/or Date Published) are straightforward in that you have asked for something quite specific. Such a search is common place to the regular library user because this information may have been available (although not nearly as quick) by using the old cards system. L.I.E.S. however has an added feature that will allow the user to retrieve information on a specified subject. It is this feature that will be explained in greater detail because it is something that needs more understanding in order to operate it efficiently.
The subject search enables the user to ask for all material on a certain subject, however, the user must know what subjects are allowable. L.I.E.S. contains a subject tree, whereby, a predetermined list of subjects is set up for the user to choose from. To view this subject tree the user has two options:

i) ask the Librarian for a copy of the subject definitions.

ii) use the Assisted search facility (for more information see section 1.1 Assisted Search).

The user has the option of getting a printout of the retrieved items. To do this the user must press <CNTL-P> on the first page of items retrieved on the screen. For more detail on the layout of the report see Figure R-01 (Report Example) given after the screen examples following.

There are four (4) simple examples following:

Example 1 - An Author search (shown in Figures S-02 to S-05)

Example 2 - A Title search (shown in Figure S-06)

Example 3 - A Subject search (shown in Figure S-07)

Example 4 - A 'combination' search (shown in Figure S-08)

(Examples of the Assisted search facility follow the introduction given in Section 1.1 Assisted Search).

These examples are for tutorial purposes and explain not only how to request a certain search but some of the screen handling and function keys that L.I.E.S. adopts.
EXAMPLE 1 - Author Search

---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8

1 L.I.E.S  CATALOGUE INFORMATION RETRIEVAL  Sep 21  09:30  11
2 12
3 13
4 14
5 15
6 16
7 17
8 18
9 19
10 110
11 111
12 (? FOR HELP) 112
13 113
14 114
15 115
16 116
17 117
18 118
19 119
20 120
21 121
22 122
23 CNTL-A : ASSIST 123
24 CNTL-E : EXIT 124

---+---1---+---2---+---3---+---4---+---5---+---6---+---7---+---8

Figure S-02: Catalogue Information Retrieval Screen Example

This is the screen that is first returned on entry to the Catalogue Information Retrieval function. To move around the screen use the <TAB> key to move from field to field and the left and right arrow keys for movements within a field.

To erase a character the user should use the delete key and if he/she wishes to erase the whole field he/she can use the <ESC> (erase screen component) key.

To position the cursor at the beginning of a field the user should press the <HOME> key.

Once the user has pressed <RETURN> the search will begin on the criteria specified.

At any time if the user requires help then they may press <CNTL-A> (ASSIST) which will give the user a brief tutorial on how to use the function.

When finished the user should press <CNTL-E> (EXIT) to exit to the L.I.E.S. logo screen.
An example of what might be a typical search is shown above. The user has requested a search of all books (specified by placing a 'B' in the medium field), published between the years of 1980 and 1982 (specified by placing those years in the published dates fields), written by Allan Abraham (specified by inputing the Author's surname and first given name in the appropriate fields).

When the user hits <RETURN> the search will be started. If no data meets the search criteria then an appropriate message will be displayed. If there are items found that meet the search criteria then they will be displayed three items at a time as seen in Figure S-04 overleaf.
EXAMPLE 1 - Author Search (Cont.)

This is an example of the screen that is returned when a number of items were found that met the search criteria.

A message is displayed to tell the user how many items were found that met the search criteria (in this case 5 items). At this stage the user may decide that he/she would like a hardcopy report of these items. To invoke a printout the user must press <CNTL-P> (PRINT). This will then send a printout of the items to the local printer. An example of a report produced can be seen in Figure R-01 at the end of the following screen examples.

It can also be seen that one of the books has 0 copies available; this implies that this book (or if there is more than one copy, all of the copies) is currently on-loan.

If the user has found the item that he/she was looking for and does not want to see the next two items then he/she must press <CNTL-E> (EXIT), which will take the user back to the initial Catalogue Information Retrieval screen. Otherwise, if the user wishes to view the next two items he/she must press the Down Arrow, which will then display the next screen, as seen in Figure S-05 overleaf.
**EXAMPLE 1 - Author Search (Cont.)**

---+---2---3---4---5---6---7---8

<table>
<thead>
<tr>
<th></th>
<th>LI.E.S</th>
<th>CATALOGUE INFORMATION</th>
<th>Sep 21 09:30</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2 ITEMS TO DISPLAY</td>
<td>ITEM NUMBER: 312 MEDIUM: B DEWEY NUMBER: 123/36.215</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>AUTHOR: ADAMS B.J. ABRAHAM A.J.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>TITLE: THE COMBINED ANALYSIS OF DATABASE DESIGN</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>PUBLISHER: PENGUIN CLASSICS, BOTANY RD, MASCOT, NSW</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>PUBLISHED: 1981 NO. COPIES AVAILABLE: 1</td>
<td></td>
</tr>
</tbody>
</table>

---+---1---2---3---4---5---6---7---8

<table>
<thead>
<tr>
<th></th>
<th>ITEM NUMBER: 314 MEDIUM: B DEWEY NUMBER: 124/36.219</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>AUTHOR: ABRAHAM A.J.</td>
</tr>
<tr>
<td>14</td>
<td>TITLE: THE LAST BOOK OF DATABASE DESIGN</td>
</tr>
<tr>
<td>16</td>
<td>PUBLISHER: PENGUIN CLASSICS, BOTANY RD, MASCOT, NSW</td>
</tr>
<tr>
<td>18</td>
<td>PUBLISHED: 1981 NO. COPIES AVAILABLE: 3</td>
</tr>
</tbody>
</table>

---+---1---2---3---4---5---6---7---8

<table>
<thead>
<tr>
<th></th>
<th>CNTL-A : ASSIST CNTL-P : PRINT CNTL-E : EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td></td>
</tr>
</tbody>
</table>

**Figure S-05 : Catalogue Information Retrieval Screen Example**

This is a screen example of the last two items to display for the original search criteria.

It can be seen that the first item appears to be written by the author Adams, but upon closer examination you can see that the book has two authors of which Abraham is one.

If the user wanted to view those items on the previous screen then he/she should press the Up Arrow. If the user were to press the Down Arrow to try and display another screen then a message will be displayed to say that there are no more items to display. What the user must do is press <CNTL-E> (EXIT) to return to the Catalogue Information Retrieval function.

If the user had pressed <CNTL-P> (PRINT) this would produce a report of the items found. A report example with the items found above can be seen in Figure R-01 following the screen examples.
EXAMPLE 2 - Title Search

---+---1---2---3---4---5---6---7---8
11 L.I.E.S CATALOGUE INFORMATION RETRIEVAL Sep 21 09:30 11
12
13
14
15
16
17
18
19
20
21
22
23
24

Figure 5-06: Catalogue Information Retrieval Screen Example

In this example the user types in the title he/she is interested in and presses <RETURN> to invoke the search. The title must be an exact match.

Because the user has not typed in a medium the search will find all items (whether books, tapes, journals, etc.) with the title requested.

If no data meets the search criteria then an appropriate message will be displayed. If there are items found that meet the search criteria then they will be displayed similarly to the way in which the items were displayed for the successful author search (in Figures 5-04 and 5-05).
**EXAMPLE 3 - Subject Search**

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOGUE INFORMATION RETRIEVAL</td>
<td>Sep 21 09:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure S-07: Catalogue Information Retrieval Screen Example**

In this example the user types in the subject(s) he/she is interested in and presses <RETURN> to invoke the search. The subject(s) must be contained on the subject reference tree.

Because the user did not input a specific medium the search will find all items (whether books, tapes, journals, etc.) related to the subjects requested.

If no data meets the search criteria then an appropriate message will be displayed. If there are items found that meet the search criteria then they will be displayed similarly to the way in which the items were displayed for the successful author search (in Figures S-04 and S-05).
Example 4 – 'Combination' Search

In this example the user has requested a search for all books written by A.J. Abraham published on, or after, 1980 on the subject of Databases.

If no data meets the search criteria then an appropriate message will be displayed. If there are items found that meet the search criteria then they will be displayed similarly to the way in which the items were displayed for the successful author search (in Figures S-04 and S-05).
<table>
<thead>
<tr>
<th>NUMBER</th>
<th>ITEM</th>
<th>MEDIUM</th>
<th>AUTHOR</th>
<th>TITLE</th>
<th>DATE</th>
<th>PUBLICATION DATA</th>
<th>ED.</th>
<th>COPIES</th>
<th>AVAILABLE</th>
<th>DEWEY NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ.</td>
<td>THE MINUTE BOOK OF D 1980 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>TANY RD, MASCOT, NSW</td>
<td>2</td>
<td>123/34.213</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ.</td>
<td>THE HUGE BOOK OF DAT 1981 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>TANY RD, MASCOT, NSW</td>
<td>0</td>
<td>124/35.341</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ.</td>
<td>THE SUMMARIZED BOOK 1981 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>TANY RD, MASCOT, NSW</td>
<td>4</td>
<td>124/36.012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>312</td>
<td>B</td>
<td>ADAMS</td>
<td>BJ.</td>
<td>THE COMBINED ANALYSIS 1981 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>TANY RD, MASCOT, NSW</td>
<td>1</td>
<td>123/36.215</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>314</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ.</td>
<td>THE LAST BOOK OF DAT 1981 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>TANY RD, MASCOT, NSW</td>
<td>3</td>
<td>124/36.219</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*** END OF REPORT ***

Figure R-01: Search Results Report Example
1.1 ASSISTED SEARCH

The Assisted Search facility guides the user through the allowable subjects until he/she finds the subject that will best suit their needs. These allowable subjects are kept in what is referred to as a 'Subject Tree'. This subject tree is set up such that the higher levels contain the broad subject categories and the lower down the tree you travel the more specific the subjects become. For example the first level of the tree may contain the subject 'DATA'. Now the user may view what subjects 'DATA' covers by looking at its children. One of its children may be 'DATABASE' for which the user is interested. On viewing its children, 'DB ADMIN' and 'DB STORAGE', the user may then decide that 'Database' was specific enough and so may return back up the tree to select 'DATABASE' as the search subject.

To invoke the Assisted Search facility the user types in a '?' in the first character of the first subject field and presses <RETURN>. This will then return the first level of subjects of the subject tree. Upon viewing the subjects in the first level the user may want to see what subjects are contained below a given subject. To view the subjects below, the user types the number given beside the subject he/she wants to traverse in the Subject Selection field and presses the 'Down arrow'. The subjects on the next level will then be displayed for the user's perusal. If the user wants to traverse the tree further he/she uses the same procedure as just described with the 'Down arrow'. When the bottom of the tree is reached for a subject area a suitable message will be displayed indicating that no further subjects exist. If the user wants to go back up the tree he/she simply presses the 'Up arrow' to return to the previous level.

When the user has finally found the subject that he/she wishes to search on there are two ways in which they can invoke the search of that particular subject:

i) Type in the number corresponding to the subject he/she wishes to search on in the Subject Selection field and press <RETURN>.

ii) If the user wishes to restrict the search by medium, for example, he/she must press <CNTL-C> (CANCEL), to cancel the Assisted Search facility and then manually input the medium and the subject to be searched on before pressing <RETURN>.

Examples of various types of searches (including use of the Assisted Search facility are given on the pages following).
To invoke the assisted search facility the user must type a '??' in the first character of the first subject field and press <RETURN>. This will then return the subjects contained on the first level of the subject tree. This can be seen in Figure S-09 overleaf.
Figure S-10: Catalogue Information Retrieval Screen Example

All of the First level subjects are displayed when the Assisted Search facility is invoked. Now the user has the option of finding books on one of the subjects shown or researching one of the subjects shown further to see if it contains a subject which would be more specific and better suit his/her needs.

To show the children of a specific subject you should type in the number corresponding to that subject in the Subject Selection field and press the Down Arrow. If however you wanted to find the books on a subject shown on the first level you should type in the number corresponding to that subject in the Subject Selection field and press <RETURN>. The search would then be submitted.

If the user had selected 7-DATA and pressed the Down Arrow then the screen shown in Figure S-11 overleaf would be returned.
The function has now displayed the subjects on the level below the subject 'DATA'. If the user wants to peruse subjects on a lower level than the one listed he/she must type in the appropriate number of that subject in the Subject Selection field and press the Down Arrow. If however the user wanted to search for items on that particular subject he/she would instead press the <RETURN> key to invoke the search.

If the user decided that this level of subjects was too specific then he/she may return to the previous level (i.e. go back up the tree) by simply pressing the Up Arrow.

The user also has the option of cancelling the assisted search facility at any time by pressing <CNTL-C> (CANCEL).
2. **LIBRARIAN FUNCTIONS**

To enter the Librarian menu type '2' on the L.I.E.S. logo screen. This will then return a screen prompting for the password needed to enter the Librarian functions. After typing in the password required hit <RETURN>; this will then return the Librarian menu ready for use. An example of the Librarian menu can be seen in Figure 5-01 below.

```
1. BORROW AN ITEM
2. RETURN AN ITEM
3. DISPLAY BORROWER INFORMATION
4. PRINT SUBJECT DEFINITIONS
5. CHANGE PASSWORD
```

Figure 5-01: Librarian Menu Screen Example

Each of the five functions seen in the Librarian menu may be selected by inputting the relevant number of the function required. (eg. if the Librarian wishes to Return an Item he/she would simply type '2'). This will then return the required function ready for use.

Each of the functions of the Librarian are outlined in more detail in the section so indicated:

2.1 Borrow an Item
2.2 Return an Item
2.3 Borrower Information Retrieval
2.4 Print Subject Definitions
2.5 Change Password

To return to the L.I.E.S. logo screen press <CNTL-E> (EXIT).
2.1 Borrow an Item

This function allows the librarian to enter information about an item being borrowed. The librarian enters:
- Borrower number
- Catalogue number
- Copy number

After the information is entered three checks are performed:
- has the borrower any fines owing
- has the borrower 10 items on loan already
- is the item currently on loan

If all checks are passed then a message will be returned indicating that the item was borrowed successfully. Otherwise an error message will be returned indicating what check was not passed.

An example screen can be seen in Figure S-02 below.

Figure S-02: Borrow an Item Screen Example

In this example Borrower 8232076 wants to borrow Catalogue Number 2345 Copy number 1. After the Librarian types in the relevant details and presses <RETURN> the function will return indicating if the item was successfully borrowed or not.

To return to the Librarian menu press <CNTL-E> (EXIT).
2.2 Return an Item

Introduction

This function allows the librarian to enter information about an item being returned. The librarian enters:

- Catalogue number
- Copy number

After the information is entered one check is performed:

- Is the item currently on loan.

An example screen can be seen in Figure S-02 below.

---+---1---2---3---4---5---6---7---8
1 L.I.E.S RETURN AN ITEM Sep 21 09:30 11
2 12
3 13
4 14
5 15
6 16
7 CATALOGUE NUMBER : 2345_-
8 17
9 COPY NUMBER : 1_-
10 18
11 19
12 10
13 11
14 12
15 13
16 14
17 15
18 16
19 17
20 18
21 19
22 20
23 CNTL-A : ASSIST CNTL-E : EXIT 12
24 ---+---1---2---3---4---5---6---7---8

Figure S-03 : Return an Item Screen Example

A message will be returned indicating that the item was returned successfully. Otherwise an appropriate error message will be returned.

To return to the Librarian menu press <CNTL-E> (EXIT).
2.3 Borrower Information Retrieval

This function will allow the Librarian to retrieve borrower data, including a list of items currently on loan for that borrower and the date they are due back. The Librarian must enter the borrower number or the borrower surname (with or without the borrower's given name/s or initials).

Because the borrower number is a unique identity the function will return all relevant data for that borrower. However, a borrower surname may not be unique, and so if there is more than one borrower with that surname the Librarian must specify what borrower he/she wants information on. Hence, if there is more than one borrower with that surname the function will return them one at a time, listing their full given name/s and address. With this information the Librarian can select which borrower he/she wants information for. When the correct borrower is displayed the Librarian must then hit <RETURN>; with the borrower number now known the function will then return all the items that borrower has on-loan and the date they are due back.

If the Librarian performs a search on borrower name and there is only one borrower in the library with that name then information on what books that borrower has on-loan will be returned immediately (i.e. without the Librarian having to hit <RETURN> again).

If any item is overdue then the date due back (on the screen) will be highlighted for the Librarian.

The information returned to the screen on the items borrowed is a condensed version; to gain a more detailed list the Librarian may request a printed list of the items by pressing <CNTL-P> (PRINT) when the items are displayed on the screen.

Two examples are given overleaf:

Example 1 - retrieval by borrower number.
Example 2 - retrieval by borrower name.

An example of the report available (see Figure R-O1) follows Example 2.
EXAMPLE 1 - retrieval by borrower number.

---+---+---+---+---+---+---+---+---+---
 1 | L.I.E.S | BORROWER INFORMATION RETRIEVAL | Sep 21 09:30
 2 | 12
 3 | 13
 4 | BORROWER : NUMBER : 8232076
 5 | SURNAME : ---------------------
 6 | GIVEN1 : ________ GIVEN2 : ________
 7 | 17
 8 | 18
 9 | 19
 10 | 20
 11 | 21
 12 | 22
 13 | 23 CNTL-A : ASSIST
 14 | 24 CNTL-E : EXIT
 15 | 25
 16 | 26
 17 | 27
 18 | 28
 19 | 29
 20 | 30
 21 | 31
 22 | 32
Figure S-04 : Borrower Information Retrieval Screen Example

In this example the Librarian types in the Borrower number and presses <RETURN>.

The screen that is returned can be seen in Figure S-05 overleaf.
**EXAMPLE 1 - retrieval by borrower number. (Cont..)**

<table>
<thead>
<tr>
<th>Borrower Information Retrieval</th>
<th>Sep 21</th>
<th>09:30</th>
</tr>
</thead>
<tbody>
<tr>
<td>BORROWER : NUMBER : 8232076</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SURNAME : BYRNESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GIVEN1 : ANDREW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GIVEN2 : JOHN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDRESS : 18/58 KEIRA ST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOLLONGONG 2500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>TITLE</th>
<th>DATE DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>THE BIG BOOK OF DATABASE DESIGN FOR BEGINNERS</td>
<td>2009/06</td>
</tr>
<tr>
<td>312</td>
<td>THE LITTLE BOOK OF DATABASE DESIGN FOR INTERMEDIATES</td>
<td>2009/06</td>
</tr>
<tr>
<td>2345</td>
<td>THE COMBINED BOOK OF DATABASE DESIGN</td>
<td>2009/06</td>
</tr>
<tr>
<td>12</td>
<td>THE RED BOOK OF CIRCUIT DESIGN</td>
<td>2009/06</td>
</tr>
<tr>
<td>657</td>
<td>HOW TO HANDLE SOFTWARE</td>
<td>06/10/86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CNTRL-A : ASSIST</th>
<th>CNTRL-P : PRINT</th>
<th>CNTRL-E : EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

Figure S-05: Borrower Information Retrieval Screen Example

This is an example of a display of the items borrower 8232076 has on-loan.

If the Librarian would like a report, with more detail than that given on the screen, then he/she must press <CNTRL-P> (PRINT). For an example of the report see Figure R-01 following Example 2.

To return to the Librarian menu press <CNTRL-E> (EXIT).
EXAMPLE 2 - retrieval by borrower name.

In this example we have the situation where John Smith would like to know what books he has on-loan and when they are due back. Because he does not know his borrower number the librarian must perform a search by the borrower name.

If there is more than one John Smith in the library then the librarian will have to firstly determine which John Smith we would like information on, but if there is only John Smith then all information will be retrieved immediately.

In this example we will assume there is more than one John Smith in the library. The screen returned can be seen in Figure 5-07 overleaf.
EXAMPLE 2 - retrieval by borrower name. (Cont.)

Because there is more than one borrower with the name John Smith the Librarian must confirm which borrower he/she requires information on. To assist the Librarian both of the borrower's given names (if there are two), the borrower's address and the borrower number is displayed.

If the borrower displayed is the borrower that the Librarian requires information on then he/she must press <RETURN> to display the items on-loan. Otherwise the Librarian must press the Down Arrow to display the next borrower with the name John Smith. When the correct borrower is displayed the Librarian must press <RETURN> to display the items that borrower has on-loan.

When the Librarian has found the borrower he/she wanted and pressed return the items on-loan will be displayed similarly to the example given in Figure S-05 for the borrower number search.

To return to the Librarian menu press <CNTL-E> (EXIT).
**BORROWER ITEMS**

<table>
<thead>
<tr>
<th>ITEM</th>
<th>MEDIUM</th>
<th>AUTHOR</th>
<th>TITLE</th>
<th>DATE</th>
<th>PUBLICATION DATA</th>
<th>COPY</th>
<th>DEWEY NUMBER</th>
<th>DATE DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ. THE BIG BOOK OF DATA 1980 PENGUIN CLASSICS, BO CASE DESIGN FOR BEGINNERS</td>
<td>25/00</td>
<td>234/123.246</td>
<td>1</td>
<td>234/123.246</td>
<td>20/09/86</td>
</tr>
<tr>
<td>312</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ. THE LITTLE BOOK OF D 1980 PENGUIN CLASSICS, BO DATABASE DESIGN FOR INTERMEDIATES</td>
<td>25/00</td>
<td>234/123.247</td>
<td>2</td>
<td>234/123.247</td>
<td>20/09/86</td>
</tr>
<tr>
<td>7312</td>
<td>B</td>
<td>ABRAHAM</td>
<td>AJ. THE COMBINED BOOK OF 1980 PENGUIN CLASSICS, BO DATABASE DESIGN</td>
<td>25/00</td>
<td>234/123.248</td>
<td>1</td>
<td>234/123.248</td>
<td>20/09/86</td>
</tr>
<tr>
<td>2345</td>
<td>B</td>
<td>ADAMS</td>
<td>BJ. THE BLUE BOOK OF CIR 1980 ENGLEWOOD CLIFFS, N. CULT DESIGN</td>
<td>25/00</td>
<td>235/125.312</td>
<td>1</td>
<td>235/125.312</td>
<td>27/09/86</td>
</tr>
<tr>
<td>12</td>
<td>B</td>
<td>ADAMS</td>
<td>BJ. THE RED BOOK OF CIRC 1980 ENGLEWOOD CLIFFS, N. CULT DESIGN</td>
<td>25/00</td>
<td>235/125.313</td>
<td>1</td>
<td>235/125.313</td>
<td>27/09/86</td>
</tr>
<tr>
<td>657</td>
<td>B</td>
<td>BENNET</td>
<td>QF. HOW TO HANDLE SOFTWARE 1980 CYNIC PRODUCTIONS, 1 REP. SMUT ST, TEMPE, NSW</td>
<td>25/00</td>
<td>269/181.069</td>
<td>4</td>
<td>269/181.069</td>
<td>06/10/86</td>
</tr>
</tbody>
</table>

*** END OF REPORT ***

Figure R-01: Borrower Information Report Example
2.4 *Print Subject Definitions*

To print the Subject definitions the Librarian should select '4' in the Librarian menu screen. This will then print the report on the local printer.

The report is formatted such that you can see what subjects 'come under' other subjects. (i.e. you can see the parent child relationships between the subjects).

An example of a portion of the report can be seen below:

<table>
<thead>
<tr>
<th>SUBJECT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARDWARE</td>
</tr>
<tr>
<td>INPUT/OUTPUT</td>
</tr>
<tr>
<td>CIRCUITS</td>
</tr>
<tr>
<td>STORAGE DEVICES</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>DATA</td>
</tr>
<tr>
<td>DATA STRUCTURES</td>
</tr>
<tr>
<td>DATA CODING</td>
</tr>
<tr>
<td>FILES</td>
</tr>
<tr>
<td>DATABASE</td>
</tr>
<tr>
<td>DB DESIGN</td>
</tr>
<tr>
<td>DB MANAGEMENT</td>
</tr>
<tr>
<td>DB SYSTEMS</td>
</tr>
<tr>
<td>DB ADMIN</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>COMPILERS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>etc.</td>
</tr>
</tbody>
</table>

From this report you can see that a first level subject might be 'DATA'. Underneath 'DATA' (i.e. 'DATA's children) are 'FILES' and 'DATABASE', for example, and underneath 'DATABASE' is 'DB DESIGN'. This is how the subject tree is structured.
2.5 Change Password

To change the Librarian password the Librarian MUST know the existing password. An example of the screen can be seen in Figure S-08 below.

```
11 L.I.E.$
21
31
41 OLD PASSWORD => ________
51
61 NEW PASSWORD => ________
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231 CNTL-A : ASSIST
241
```

Figure S-08 : Change Librarian Password Screen Example

To change the password the Librarian must type in both the existing password and the new password and press <RETURN>. After checking that the Librarian has entered the existing password correctly the function will then request that the Librarian re-type the new password to ensure that it was typed correctly in the first place. After re-typing the new password and pressing <RETURN> the password will be changed and a message returned indicating this.

To return to the Librarian menu press <CNTL-E> (EXIT).
3. LIBRARY MANAGER FUNCTIONS

To enter the Library Manager menu type '1' on the L.I.E.S. logo screen. This will then return a screen prompting for the password needed to enter the Library Manager functions. After typing in the password required hit <RETURN>; this will then return the Library Manager menu ready for use. An example of the Library Manager menu can be seen in Figure 5-01 below.

```
ENTER SELECTION ==> 1
1. MAINTAIN CATALOGUE
2. MAINTAIN BORROWER
3. MAINTAIN SUBJECTS
4. SUBMIT FILE CHANGES
5. MANAGEMENT REPORTS
6. CHANGE PASSWORD
```

Figure 5-01: Library Manager Menu Screen Example

Each of the six functions seen in the Library Manager menu may be selected by inputting the relevant number of the function required. (eg. if the Library Manager wishes to Maintain Catalogue Data he/she would simply type '1'). This will then return the required function ready for use.

Each of the functions of the Library Manager are outlined in more detail in the section so indicated:

3.1 Maintain Catalogue
3.2 Maintain Borrower
3.3 Maintain Subjects
3.4 Submit File Changes
3.5 Management Reports
3.6 Change Password

To return to the L.I.E.S. logo screen press <CNTL-E> (EXIT).
1.1 Maintain Catalogue Data

This function enables the Library Manager to Maintain Catalogue Information and may be accessed from the Library Manager menu by selecting '1'.

When the Maintain Catalogue Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Catalogue Record.

To display a catalogue record type in the catalogue number to be displayed and press <RETURN>.

To create a catalogue record place a 'C' in the Transaction field, enter the relevant information for that catalogue record, and press <RETURN>. If the catalogue record does not already exist and all information has been entered correctly then a message will be displayed indicating that the record was created successfully; otherwise an appropriate error message will be displayed.

Before a catalogue record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'U' in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a catalogue record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'D' in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.

When a record is either created, updated or deleted the transaction is written to the Catalogue Transaction File for processing at a later date. The transactions may be written to the Catalogue file by Submitting File Changes (see Section 3.4 for more detail).

An example of an Update can be seen in Figures S-02 to S-03 overleaf. (This will firstly involve a display. Examples of creates and deletes are not given because they work on much the same principles as updates and you should get an idea of how they will work through following the update example).

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTL-E> (EXIT).
The Library Manager has displayed catalogue number 2121 by typing 2121 in the catalogue number field and pressing <RETURN>. Now he/she wishes to change the number of copies to 2 and so places a 'U' in the Transaction field, overtypes the copies field (which currently shows '1') with 2 and presses <RETURN>. The screen returned can be seen in Figure S-03 overleaf.
The change has been made and a message has been displayed indicating that the update has been made successfully.

To return to the Library Manager menu type <CNTL-E> (EXIT).
3.2 Maintain Borrower Data

This function enables the Library Manager to Maintain Borrower Information and may be accessed from the Library Manager menu by selecting ‘2’.

When the Maintain Borrower Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Borrower Record.

To display a borrower record type in the borrower number to be displayed and press <RETURN>.

To create a borrower record place a ‘C’ in the Transaction field, enter the relevant information for that borrower record, and press <RETURN>. If the borrower record does not already exist and all information has been entered correctly then a message will be displayed indicating that the record was created successfully; otherwise an appropriate error message will be displayed.

Before a borrower record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a ‘U’ in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a borrower record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a ‘D’ in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.

When a record is either created, updated or deleted the transaction is written to the Borrower Transaction File for processing at a later date. The transactions may be written to the Borrower file by Submitting File Changes (see Section 3.4 for more detail).

An example of an Update can be seen in Figures S-04 to S-05 overleaf. (This will firstly involve a display. Examples of creates and deletes are not given because they work on much the same principles as updates and you should get an idea of how they will work through following the update example).

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTL-E> (EXIT).
The Library Manager has displayed borrower number 8231561 by typing 8231561 in the borrower number field and pressing <RETURN>. Now he/she wishes to change the address for that borrower so places a 'U' in the Transaction field, overtypes the address fields with the new address and presses <RETURN>. The screen returned can be seen in Figure 5-05 overleaf.
Figure 5-05: Maintain Borrower Data Screen Example

The change has been made and a message has been displayed indicating that the update has been made successfully.

To return to the Library Manager menu type <CNTL-E> (EXIT).
3.3 Maintain Subject Reference Data

The Subject Reference File is kept in what is referred to as a 'Subject Tree'. This subject tree is set up such that the higher levels contain the broad subject categories and the lower down the tree you travel the more specific the subjects become. For example the first level of the tree may contain the subject 'DATA'. Now below 'DATA' may be the subjects 'DATA STRUCTURES', 'DATA CODING', 'FILES', 'DATABASE' and 'DATA STORAGE'. Then below 'DATABASE' may be the subjects 'DB DESIGN', 'DB MANAGEMENT', 'DB SYSTEMS' and 'DB ADMIN'. In this example the subject 'DATA' is the parent of the subject 'DATABASE', and 'DATABASE' is the parent of the subject 'DB ADMIN'. This also implies that the subject 'DB ADMIN' is a child of 'DATABASE', and 'DATABASE' is a child of the subject 'DATA'.

The 'root' of the tree is the highest level of the tree and will not have a parent subject of course; but this subject has been previously set up in creating the tree. We assume that the subject 'COMPUTING' will always be the 'root' of the tree and so any subject added will have a parent.

This function enables the Library Manager to Maintain Subject Information and may be accessed from the Library Manager menu by selecting '3'.

When the Maintain Subject Data screen is returned the user has the facility to either:

Display, Create, Update, or Delete a Subject Record.

To display a subject record type in the Subject to be displayed and press <RETURN>.

To create a subject record place a 'C' in the Transaction field, enter the new subject, the parent of that subject, the description of the new subject and press <RETURN>. If the subject does not already exist and the parent subject does exist then a message will be displayed indicating that the subject was created successfully; otherwise an appropriate error message will be displayed. (If the subject being created is to be a first level subject then its parent is 'computing').

Before a subject record can be updated it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'U' in the Transaction field, make the relevant changes and press <RETURN>. If the updates were acceptable then a message will be displayed indicating that the record was updated successfully; otherwise an appropriate error message will be displayed.

Before a subject record can be deleted it must be displayed. This ensures that the correct record will be processed. After the record has been displayed type a 'D' in the Transaction field and press <RETURN>. A message will then be displayed indicating that the record was deleted successfully; otherwise an appropriate error message will be displayed.
When a subject is either created, updated or deleted the subject reference file is updated immediately.

An example of an update can be seen in Figure S-06 below. (This will firstly involve a display. Examples of creates and deletes are not given because they work on much the same principles as updates and you should get an idea of how they will work through following the update example).

If at any time during an update the Library Manager decides to change his/her mind then he/she may press <CNTRL-C> (CANCEL) to cancel any changes that may have been typed in.

To return to the Library Manager menu press <CNTRL-E> (EXIT).

---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
1 | L.I.E.S MAINTAIN SUBJECT REFERENCE DATA Sep 21 09:30 11
2 | 12
3 | 13
4 | TRANSACTION : _ (blank:Display, C:Create, U:Update, D:Delete ) 14
5 | 15
6 | SUBJECT : DATABASE PARENT : DATA 16
7 | 17
8 | DESCRIPTION : 18
9 | 19
10 | 110
11 | 111
12 | 112
13 | 113
14 | 114
15 | 115
16 | 116
17 | 117
18 | 118
19 | 119
20 | 120
21 | 121
22 | 122
23 | CNTL-A : ASSIST CNTL-C : CANCEL CNTL-E : EXIT 123
24 | 124

---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

Figure S-06: Maintain Subject Reference File Screen Example

The Library Manager has typed in the subject 'DATABASE' and pressed <RETURN> to display the subject and its parent. If the Library Manager wanted to add a description he/she would simply place a 'U' in the Transaction field, type in the description and press <RETURN>. If the update was OK then a message will be returned to the screen indicating that the updates were performed successfully; otherwise, an appropriate error message will be displayed.
3.4 Submit File Changes

This menu type function enables the Library Manager to submit file changes to update the Catalogue, Borrower or On-Loan transaction files.

The user has the option to request that one file be updated or he/she may request that all files be updated.

The On-Loan transaction file contains the days borrowing and returns transactions. This will be run upon request, which is assumed will be a nightly task.

The Borrower transaction file contains the transactions for new, updated or deleted borrowers since the last time the Borrower changes were submitted. This will be run upon request, which is assumed will be on the night following a new borrower being added.

The Catalogue transaction file contains the transactions for new, updated or deleted catalogue records since the last time the Catalogue changes were submitted. This will be run upon request, which is assumed will be once every six months or by special request.

*** WARNING *** - this batch job will start running at midnight and will render the system unavailable for the duration of the update.

An example of the Submit File Changes screen can be seen in Figure 5-07 overleaf.
To select the update required input the appropriate selection (i.e. 1, 2, 3 or A). This will then submit a batch job to update the relevant files.

To return to the Library Manager menu press <CNTL-E> (EXIT).
3.5 Management Report Generator

This menu type function enables the user to select any one of the management reports offered by the system. These reports are:

1. CATALOGUE ENTRY REPORT - This is a report of all the items currently on the CATALOGUE file.
   (An example of this report can be seen in Figure R-U1)

2. BORROWER REPORT - This is a report of all the borrowers that are currently members of the library.
   (An example of this report can be seen in Figure R-U2)

3. CURRENT ON LOAN REPORT - This is a report of all the items currently on loan.
   (An example of this report can be seen in Figure R-U3)

4. CURRENT OVERDUE REPORT - This is a report of all items that are overdue.
   (An example of this report can be seen in Figure R-U4)

To select the report required the user must input the appropriate selection (i.e. 1,2,3, or 4). This will then submit the report and print it at the local printer.

If there is no data found for the particular report, an empty report will be written to notify the user of this.

An example of the screen from which the reports are submitted can be seen in Figure S-08 below.
To select the report required input the appropriate selection (i.e., 1, 2, 3, or 4). A message will be returned indicating that the report was submitted successfully and the report will print out at the local printer.

If there is no data found for the particular report, an empty report will be written to notify the user of this.

To return to the Library Manager menu press <CNTL-E> (EXIT).
<table>
<thead>
<tr>
<th>ITEM NUMBER</th>
<th>MEDIUM</th>
<th>AUTHOR</th>
<th>TITLE</th>
<th>PUB YEAR</th>
<th>PUBLISHER NAME</th>
<th>PUB ED.</th>
<th>NUMBER</th>
<th>COPIES</th>
<th>COPIES BORROWED</th>
<th>COPIES RESERVED</th>
<th>SUBJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>BENNET</td>
<td>INTRODUCTION TO COMP UTER APPLICATIONS</td>
<td>1976</td>
<td>J. PRENTICE-HALL</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>APPLICATION</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>ABRAHAM</td>
<td>THE MINUTE BOOK OF D B DESIGN AND ADMIN</td>
<td>1978</td>
<td>PENGUIN CLASSICS, BO TANY RD, MASCOT, NSW</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>DB DESIGN</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>SHELTON</td>
<td>THE HUGE BOOK OF DAT ABASE DESIGN</td>
<td>1980</td>
<td>ENGLEWOOD CLIFFS, N. J. PRENTICE-HALL</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>DATABASE</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>ISAACS</td>
<td>THE COMBINED BOOK OF 1981 CIRCUIT THEORY</td>
<td>1981</td>
<td>COLLINS PRINTERS, VA MCOUVER</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td>CIRCUITS</td>
</tr>
</tbody>
</table>

*** END OF REPORT ***

Figure R-01: Catalogue Report Example
<table>
<thead>
<tr>
<th>BORROWER NUMBER</th>
<th>BORROWER NAME</th>
<th>ADDRESS</th>
<th>TELEPHONE NUMBER</th>
<th>ITEMS ON LOAN</th>
<th>COPY</th>
<th>FINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>8232076</td>
<td>BYRNEs</td>
<td>18/58 KEIRA ST WOLLONGONG</td>
<td>262695</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8232189</td>
<td>HIRO</td>
<td>214 BOURKE RD BERKELY</td>
<td>743220</td>
<td>482</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8234328</td>
<td>BARKER</td>
<td>429 UPPER CLASS RD LAKE HEIGHTS</td>
<td>742713</td>
<td>321</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8239358</td>
<td>BOW</td>
<td>4 YURUGA AVE HILLYVILLE</td>
<td>272620</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8239923</td>
<td>GROSSMAN</td>
<td>12 JIMMYBLACK ST BACKABOURKE</td>
<td>272929</td>
<td>28</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

*** END OF REPORT ***
<table>
<thead>
<tr>
<th>NUMBER</th>
<th>ITEM</th>
<th>NUMBER</th>
<th>COPY</th>
<th>BORROWER</th>
<th>DATE</th>
<th>DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>1</td>
<td>8239923</td>
<td>17/09/86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.321</td>
<td>1</td>
<td>8234328</td>
<td>25/09/86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.482</td>
<td>1</td>
<td>8232189</td>
<td>19/09/86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.3291</td>
<td>3</td>
<td>8232189</td>
<td>27/09/86</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*** END OF REPORT ***

Figure R-03: On-Loan Report Example
**CURRENT OVERDUE REPORT**

<table>
<thead>
<tr>
<th>NUMBER ITEM</th>
<th>NUMBER COPY</th>
<th>NUMBER BORROWER</th>
<th>DATE DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>1</td>
<td>8239358</td>
<td>19/09/86</td>
</tr>
<tr>
<td>482</td>
<td>1</td>
<td>8232189</td>
<td>19/09/86</td>
</tr>
</tbody>
</table>

*** END OF REPORT ***

Figure R-D4: Overdue Report Example
3.6 Change Password

To change the Library Manager password the Library Manager MUST know the existing password. An example of the screen can be seen in Figure S-09 below.

```
11 L.I.E.S Sep 21 09:30 11
21 12
31 13
41 OLD PASSWORD ==> ________
51 14
61 NEW PASSWORD ==> ________
71 15
81 16
91 17
101 18
111 19
121 110
131 111
141 112
151 113
161 114
171 115
181 116
191 117
201 118
211 119
221 120
231 CNTL-A : ASSIST CNTL-E : EXIT 121
241 122
```

Figure S-09: Change Library Manager Password Screen Example

To change the password the Library Manager must type in both the existing password and the new password and press <RETURN>. After checking that the Library Manager has entered the existing password correctly the function will then request that the Library Manager re-type the new password to ensure that it was typed correctly in the first place. After re-typing the new password and pressing <RETURN> the password will be changed and a message returned indicating this.

To return to the Library Manager menu press <CNTL-E> (EXIT).