1987

Laboratory notebook tracking system /

Deborah Louise Rosche
Lehigh University

LABORATORY NOTEBOOK TRACKING SYSTEM

by

Deborah Louise Rosche'

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the degree of

Master of Science

in

Industrial Engineering

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 4, 1987

Date

John R. [illegible]

Professor in Charge

S. E. Kane

Chairman of the Department

Table of Contents

Abstract 1

The Need for Change 3

Deficiencies of the Current System 5

Other Lab Notebook Tracking Systems 6

Other Commercial Systems 9

Datalib 11

Datalib Data & File Structure 13

Cost/Benefit Analysis 17

File Conversion Background 19

Data File Conversion 20

Bibliography 57

Biographical Sketch 59

List of Figures

Figure 1 Sort by Notebook Number for All Notebooks 25

Figure 2 Sort by Current Holder for All Active Books 26

Figure 3 Sort by Original Assignee for All Active Books 27

Figure 4 Sort by Notebook Number for Reassignment History 28

Figure 5 Data Entry Update Form 29

Figure 6 Data Entry History Form 30

Figure 7 Current File Structure 31

Figure 8 TBMS Features Summary 32

Figure 9 Datalib DDF 33

Figure 10 Datalib Completed DDF Record 35

Figure 11 Datalib File Structure 36

Figure 12 Datalib Online Record 37

Figure 13 Cost/Benefit Analysis 38

Figure 14 Conversion Program Scheme 39

Figure 15 Load Diagnostics 40

Figure 16 Datalib Search by Record Type 41

Figure 17 Datalib Search 42

Figure 18 Datalib Search & Record Change 43

List of Appendixes

Appendix A Analysis of Software Packages 44

Appendix B Space Insertion Program 50

Appendix C Conversion Program 51

ABSTRACT

The physical conversion of data from one environment to another is a complex job. When Information Services (IS) decided there was a real need to convert the laboratory notebook tracking system from the mainframe environment to the VAX environment because of untimely batch reports, record structure restrictions, limited reports and lack of support for the Asi-st© programming, the usual obstacles were encountered: little documentation and no one who really understood the current system. The data was to be transferred from a batch, fixed record, sorted report system to an indexed, variable record, interactive system. The problems included converting a standard ASCII file to a text file with field tags, combining history records with appropriate lab records into a single record per notebook number and removing tabs from the original data and inserting spaces.

IS had recently installed Datalib software on the VAX to run the online card catalog. Datalib provided the indexing, updating, searching and security features that were needed for the lab notebooks. It was decided to add the lab notebook file on Datalib.

The conversions were performed with two programs. The first program located tabs in each record, replaced them with a space and inserted the necessary number of spaces.

The second program consisted of three sections. The file was

read and lab data written to a temporary file in the sequence which it would later be needed. An index file was created and the record number and notebook number written to it. A history file was also created, and eight Z's, (the length of the notebook number field), were written to the file for each record. The original file was then read again. When a history record was located, the index file was checked to determine the record number of the corresponding lab record. The history data was then written to the history file in that record, overwriting the Z's. An additional file had to be created to record the notebook numbers of those history records that did not have lab records. Because of an input error in the original data, there were two such history records. The temporary and history file were then read. If the history record contained only Z's, there was no history data. The records were written to the final file with field tags and record flags added.

The reformatted file was then uploaded to Datalib using its internal loading and indexing utilities. A diagnostic log file was created during the load, identifying any problems and verifying the number of records loaded.

The Need For Change

Information Services Department of Corporate Research Services has always held the custodial responsibility for research laboratory notebooks.

An Air Products' Standard Practice defines original research records as:

- "Original research records are permanent, documented records of original research and development data and discussions relative to conceptions, experimentations, observations, progress and results of research work projects."

This Standard Practice also defines laboratory notebooks as:

- "Lab notebooks are the preferred medium for recording original research data and should be used whenever practical."

The purpose of the Standard Practice is described as:

- "Establishing the requirements concerning the maintenance, witnessing, and safeguarding of original research records. These records must be properly maintained and readily accessible for use in subsequent research, to establish the origins of proprietary information, legal proceedings, and to protect the Company's interests."

Air Products' Laboratory Notebook Tracking System was developed to fulfill this purpose. It was designed with the following objectives:

- To organize data in records maintained at several locations at Air Products and in records obtained through acquisitions over the years.

- To maintain a centralized record of all notebooks.

- **To provide timely information on the status and location of all laboratory notebooks.**

- **To generate special reports, for example, lists of notebooks due to be microfilmed, in order to meet standard practice requirements.**

The tracking system was developed to provide information on assignment, status and location of notebooks, but not subject content. The Air Products' Laboratory Notebook Tracking System is currently run on the IBM 3090 mainframe by a program run in Asi-st©, basically a reporting language. Asi-st© accesses the file to be used and retrieves the required data to generate the required reports.

The lab notebook record contains original assignment data - notebook number, assignee name, employee number, assignment date, department, completion date, current holder and microfilming information. Currently received monthly reports are sorted by:

- Notebook number for all notebooks (Figure 1).
- Current holder for all active books (Figure 2).
- Original assignee for all active books (Figure 3).

Every six months, approximately, a report is requested sorted by original assignee for all notebooks.

A separate record is maintained for notebook history data. If a notebook is reassigned to another person, a history record is generated to keep track of the book. Notebook number, user name and return date are recorded. The current system allows four reassignments. If a fifth should occur, the oldest reassignment is dropped. Monthly reports are sorted by notebook number giving all reassignment history for that book (Figure 4).

As of February 1987, there were 11,655 master lab notebook records and 1,515 history records.

New data is currently added to the system in the following manner: A data entry form is filled out for each notebook as it is signed out (Figure 5). If a notebook is reassigned, a data entry form for history is filled out (Figure 6). The forms are submitted for a batch run at the end of each month. It is necessary to proofread the new data, and any corrections needed are then resubmitted with the next month's run. An average of two hundred changes, including new assignments, reassignments, recalls and microfilming, are processed in this manner every month. Because of the time involved, only one printout is updated with notations during the monthly interim, between update processing times. By the time the new listings are generated, the newly added data can already be as much as one and one-half months old. This makes it more difficult to collect up-to-date information when needed on an individual or a notebook.

Deficiencies of the Current System

There are several problems with the current system. Fields are fixed length. The assignee field, for example, is only eleven characters long, not long enough for the last name and initials of numerous employees. All updating is done in batch mode. With the large number of changes, the paperwork and proofreading required is very time consuming. Also, because of batch updating, the data can be as much as one and one-half months old by the time it gets into the system. Because of interim manual updating on the print-out, there are several places that have to be checked when gathering information on individuals or books. Only the aforementioned five reports are easily generated. Additional Asi-st© programming would be required if special reports or a change in reports were needed. Asi-st© is no longer used in new programming at Air Products. If major revisions were needed or Asi-st© were removed, Information Services would have to pay for the conversion. The current file structure is shown in Figure 7.

Other Lab Notebook Tracking Systems

Rohm & Haas Company Research Libraries in Springhouse, PA currently has their lab notebook system on the System 2000[1], a general purpose database management system run on computers such as an IBM 360, Univac 1100 or CDC 6000. It has both batch and interactive update and report capabilities. Purchased in the

[1] Pheobe Rosenberry and Barbara G. Prewitt, Applications of System 2000 Database Management Software at Rohm & Haas Company Research Libraries, The Information Age in Perspective, Proceedings of the 41st ASIS Annual Meeting, XV, (New York, 1978), pp. 274-277.

early 70's, the lab notebook information is in the book circulation database. The file has a heirarchical structure. It includes information such as notebook number, person signed out to, assignment date, department number, whether the book is opened or closed, microfilm number and loan information. There is no subject indexing. Information can be retrieved by notebook number, author, current holder or date. Reports are generated every six months on books on loan or open for more than one year for recall. The library, however, presently feels that this is not the best system for this type of information. They are currently investigating Datalib for their book and journal circulation, and may put the lab notebook information on at a later time.

The Technical Information Center (TIC), S.E.L. Computer System Division of Gould Inc., designed and wrote their own programs, in Fortran 77+, for their database[2]. The programs run on a 32-bit minicomputer, the GOULD 32/77. It has two megabytes of MOS memory and two 300 megabyte disks. The system contains files for book and journal collection, component specifications used for design and purchasing, publication orders, copyright records, interlibrary loans, competitive manuals, online

[2] L. Susan Hayes and Judith A. Vogel, "IMP-32 - An Interactive Management Program", <u>Database</u>, V, No. 3, (August 1982) 56-62.

card catalog and engineering notebooks. Updating and searching are interactive. The engineering notebook file contains book number, author(s), date assigned, a user assigned description, and title. All the keys are searchable. However, TIC is currently in the process of converting the engineering notebook file because of lack of support for the present system. They plan to place the file on an IBM-PC in DBASEIII and add loan history data, bar code labels and Marc format.

RCA in Princeton, NJ, has their union list on Datalib, but their lab notebook system is run on an IDM 500, a BDMC[3] (backend data base machine). It is a relational database with two relations in table form. The first table contains the employee name, employee number, where the notebook is located and any comments. The second table contains employee number, notebook number, and dates assigned and completed. If the employee number is not known, the first table is searched and a join performed to find the desired record in the second table. The database contains a one to many relationship - employee number to notebook. The sessions are interactive. Plans have been made to put the database on Datalib, but it has not yet been done.

Allergan Pharmaceuticals, in California, and Cibbarelli and Associates designed an automated retrieval system, LIBNOTES, for

[3] Personal communication with Howard Hoffman, Programmer, RCA.

their lab notebooks in the late 1970's.[4] The software, written in Fortran, provided seven access points to the notebook data: notebook number, person assigned to, department, inclusive years of experiments, project numbers, formulation numbers and substance names. Controlled vocabulary and cross references were used. It offered unlimited field lengths and a choice of listing formats. Sorted lists provided a basic reference to the notebooks. Problems requiring more complex searching, e.g. using Boolean operators, were done in batch searches.

About seven years ago, they converted their system to Inquire, a DBMS by Infodata.[5] Written in PL/1 and assembler, it runs on their IBM mainframe and has a VSAM file structure. It can accommodate up to 1,000 fields with a total record length of 32,000 characters. In addition to the fields previously used, current location and microfilming date are included. Input is interactive, using a full screen data entry display. Searching and sorting are also interactive using any field.

Other Commercial Systems

Ten commercially available text-based management systems

[4] Pamela Cibbarelli, Carol Tenopir and Retha Ott, LIBNOTES, A Laboratory Notebook Retireval System, The Information Age in Perspective, pp. 67-69.
[5] Personal communication with Allergan.

(TBMS) for PCs were compared in a recent issue of PC Magazine.[6] They are a combination of word processor and database with full-screen editors, variable field and record lengths, and retrieval features. Price, file structure, editor features, data import-export and manipulation, searching output, functions, command strategy and special features were compared for Executive Filer, Instant Recall, SquareNote, Sequitur, Notebook II, AskSAM, DayFlo, Mist, and Inmagic and Marcon Plus (Figure 8). Datalib compares favorably to these TBMS.

Datalib does require that fields be defined beforehand, but prompts the user for data entry. Only those fields that contain data need be shown in the output. Fields can be defined as searchable or display only, so there can be as few or as many access points as needed. If the possiblity for multiple entry exists, or if a field will be longer than 256 characters, that field can be defined as repeating. There is no maximum record size.

Records can be edited individually or global changes can be made in the authority files. These changes are automatically posted to the appropriate records. Indexing in Datalib is interactive, the information is searchable as soon as it is added

[6] Vincent Puglia, "TBMS Database Power Unleashed," PC Magazine, (November 1986), 211.

or changed. All keywords in fields defined as searchable are indexed. Searching can be done on strings or keywords, where defined. Boolean, date and proximity searching is supported. Searching can be either command or menu driven. The first letters of a command can be used instead of the full word if it is unique. Datalib provides a list of stop words. Sorting of output should be available with the next release. The file is in ASCII format, but because of indexing and keying, the data does need to be massaged before direct import.

Datalib supplies various standard stored report definitions. Custom reports or headers can be added. Query definitions can also be stored to rerun at a later time. Security access, online help messages, and copy protection are also supported.

The major advantage of Datalib is the maximum file size. IS is currently on a 550 megabyte disk. If at some point in time the disk was filled, another would simply be added.

Datalib

Datalib software is a database management and information retrieval software package, sold by Sigma Data Services Corp.[7] Written in ANSI standard COBOL[8], it runs on the R&D Digital VAX

[7] Datalib product summary, Data Sigma Corporation.

[8] Audry N.Grosch, Minicomputers in Libraries, 1981-82, the Era of Distributed Systems (Knowledge Industry Publishers, 1982), pp. 84-85.

minicomputer. The software supports automation of several library operations, including cataloging of library resources, circulation of resources, records and accounting for library acquisition and serials management. Datalib was chosen for use at Air Products after a review of several software packages, including a review of Datalib, Techlib, Bibliotech, Ulisys and Atlas (Appendix A). The cataloging, retrieval, circulation, authority control, acquisitions and serials functions, vendor support and technical environment of the packages were compared. Datalib was chosen for several reasons. It provides flexibility in defining record formats and allows the user to define the record format as well as which fields in the record will be searchable. Fields can be up to 256 characters in length. It provides online updating of information, contains an efficient data structure and provides outstanding retrieval capabilities. Data elements can be indexed by full key and/or key words. Records are indexed as they are added for immediate retrieval. Retrieval can be done using any field, as defined by the customer. Truncation, boolean operators and proximity can be used.

Datalib provides access security at several levels - user identification, file, record and element. The user profile specifies the access level and what functions (e.g. editing, deleting, adding, reading) that user has. Similar to the consultants file, the lab notebook file will be accessible only to the appropriate IS staff for both read and write capabilities.

Datalib creates a log file of transactions which can be used to recreate records generated between regular backups. As part of VAX's regular maintenance, the files will be backed up nightly. An image of the entire system is done weekly. The bibliographic function (cataloging and retrieval) will accept private as well as public files that have been converted to Datalib format.

Datalib is currently being used to manage and provide access to a variety of materials, including books, material safety data sheets, subscriptions, consultants and Air Products' patents. It is, therefore, a logical choice for the lab notebook tracking system. There are several additional advantages to having the lab notebook tracking system on Datalib. Because of the online updating and immediate retrieval, the system will provide timely information. For these same reasons, it will greatly reduce maintenance time. Statistics that now have to be kept manually can be retrieved online, such as how many changes or new additions have occurred for a specific time period. Notebooks are recalled on a regular basis for microfilming; a list can be generated online of those notebooks that need to be recalled.

Datalib Data & File Structure

Datalib consists of a single database composed of files for each customer. The content, format and other characteristics of records in the database are defined in the Data Definition File (DDF). The DDF defines the record types by name, data type, attributes and a list of the elements and nested records which make up the record type. The DDF is shown in Figure 9.

Within the DDF, it is defined in the following order: first, lists of codes with print names are defined for later use by elements defined as coded. Elements to be included in authority records and the authority record itself are defined next. (There are no authority records for the record type Lab_Notebooks). Nested and master record elements are then defined in the order: element name, type, options, attributes. Nested records are defined in the order: record name, type, options, attributes, previously defined elements and nested records. Reader comments may be added to the DDF by prefixing them with an exclamation point. Record and element names must begin in column one in their definition. Only those items in the DDF which pertain to the Lab_Notebooks record will be described. The following elements and their element names are included in the Lab_Notebooks record:

notebook number	nbr_book
original assignee	orig_assign
employee number	emp_nbr
department	dept
assignment date	date_assign
completion date	date_comp
microfilm date	date_mic
microfilm number	mic_nbr
witness	wit
location	location
notes	notes
library	library

The history section of the record includes:

assignment name	assignment
employee number	emp_nbr
loan date	loan_date
return date	return_date

Lists are specified at the beginning of a section of the DDF before the elements are defined by name and data type (LIST). Lists of valid codes and optional print names must be defined for CODED elements. In the Lab_Notebooks record, the WIT and LIBRARY elements contain lists. The WIT list, YNLIST, is simply Y for yes or N for no. The LIBRARY list, LIBLIST, contains the accepted library locations. This list is also used elsewhere in the database.

The element is a single item of data which, once defined, can be used in multiple record types and files. They are defined by name, the type of data, options and attributes. Element names begin in column one and must be unique. If the name contains two words they must be connected, for example, with an underscore or period. Element names can be up to sixteen characters in length.

Data type follows the element name and is a mandatory designation. The Lab_Notebooks record contains ALPHA, CODED and DATE data types. ALPHA is any alphanumeric or ASCII symbol. CODED data contains a code from a list of valid codes as previously defined in a LIST. DATES entered in the format, for example, Dec 1986, December 1986, 12/86, Dec/1986, December 1 1986 or 12/1/86 will be converted to Dec 1986 or Dec 1 1986.

Options are true/false conditions specifying how data will be entered or retrieved. The value of an option is true if it is listed in the element definition, otherwise, it is false. Options can be specified for a particular record type or for that element in all record types. Four element options are used in the Lab_Notebooks record: KEYED, MASSTEXT, TEXT and UNIQUE. Data specified as KEYED is indexed for retrieval or browsing as the full data string, e.g. the full title. Data in elements specified as TEXT are indexed as individual words for retrieval or browsing. Elements defined as MASSTEXT are treated as units of data when entered and displayed. This option is useful for abstracts or notes. Data entered for an element defined as UNIQUE must be unique for all record types which include the element. These elements must also be KEYED. Attributes are optional conditions with the value defined in the format. Lab_Notebooks record only uses the LIST attribute. LIST provides the name of the previously defined list of codes for CODED elements.

Records, a collection of related data, are defined by name, data type, attributes, previously defined elements and nested records. The record name must be unique from any other record or element names. Naming rules are the same as for elements. The data type is RECORD. The CLASS attribute maximizes searching response time by defining which bibliographic record types can be searched together. Previously defined elements and nested records are listed in the order in which the system will prompt for them when a record is created. The Lab_Notebooks record has the additional options of mandatory and repeating. Data must be entered for an element designated as mandatory. Repeating elements allow for multiple values to be entered.

Nested records, records within records, are defined by NAME, RECORD, attributes and elements. These subsets of elements repeat as a unit within a record. Datalib allows for three nested levels. The completed record is shown in Figure 10.

Datalib file structure for this record is shown in Figure 11. Each record begins with the record type and ends with $END. The history borrow record repeats within the master record and is in the third nested level. The online record is shown in Figure 12.

Cost/Benefit Analysis

The factors used in the cost/benefit analysis are shown in Figure 13. Costs for this project are minimized for several reasons. Because the software is already in use for another function, there are no additional software or hardware costs. Vendor support and updates are part of a previous agreement. Currently, Information Services pays a flat annual fee for VAX usage, which will not change when another file is added. Terminals from which this system can be used are already in place. The only costs are a one-time conversion cost, estimated at seventy hours. A MIS labor charge of $50/hour is used.

Personnel training for the new file is estimated at two hours. Minimal training will be needed, as the system that the file will be on is currently operational.

Benefits were calculated for a five year system life expectancy. An estimated 22.8 hours per year will be saved in the elimination of manual labors such as filling out data entry forms, making microfiche entries, transferring entries and proofreading. Twenty-four hours per year will be saved with faster processing of information and interactive updating. Reducing the volume of paper handled will save 13.2 hours per year with the elimination of having to make interim entries of the printout. The current yearly cost for storing and running the program on the mainframe, $2,500, will be saved when the file is on the VAX.

Intangible benefits such as smoothing of operational flows, rise in service quality and performance, future expansion capabilities, more efficient use of personnel and ability to compile statistics will be realized. Perhaps the most important intangible benefit is the increased control of proprietary information. The value of this system to the company lies in being able to track and control the whereabouts of issued lab notebooks. A survey of the R&D community provided several examples of time and money saved in research and legal fees by being able to locate needed lab notebooks. Completed notebooks are often referenced as to lab procedures, sample numbers, recipes and test results. In one case, a notebook that contained proprietary information on new research was thought to be lost or sold. Had that notebook not been able to be located, we would have lost our position for a future market. If the notebook had not been found, the research would not have been repeated, which again would have resulted in a lost market. In another instance, several notebooks involved in a legal case could not be located. Several hundred thousand dollars in legal and lab fees might have been saved if the notebooks could have been found. The new system will provide better tracking and more accurate, up-to-date information on the current location of the notebooks.

The costs of this conversion total $3,600; the benefits over five years total $17,000.

File Conversion Background

File conversion problems and techniques seem well covered in the literature. However, upon closer inspection, most of the literature discusses program software conversion for converting applications from the file to database environment. The design of the database itself, and software considerations such as program conversion, query processing, access methods and mapping to construct the database record are discussed in detail. Actual physical restructuring of the data file for loading to the database is only briefly mentioned as a complex task that has to be done, but no detailed discussions are given. Datalib is complete with programs for query processing, updating and security. The application already exists and is in use. The main problem is restructuring the data into a format that Datalib can read.

Data File Conversion

When trial data was transferred from the mainframe to the VAX, it was discovered that the VAX was 'reading' some characters as tabs. Apparently, tabs were used in data entry, although the mainframe does not read them as tabs. Before a conversion program could be run, spaces needed to be inserted for the tabs, and the tabs deleted. The input file was run through a program, Deltab.for (Appendix B), to add the needed spaces. Because a tab conversion function is not currently available, this program will be added to the VAX's subroutine library.

Datalib's internal program to read and load data files treats the files as if dumped from Datalib, e.g., as in a reload. The dump file structure is that previously presented (Figure 11). The current file structure (Figure 7) has to be restructured to that of Datalib. Beginning and end of record flags and field tags needed to be added. It was decided to combine the master (lab) and the history (his) data into one record. Then, when any one lab notebook record is searched, the complete record is retrieved. Fields for employee number and loan date were added to the history level of the record, although the present file does not contain any information in these fields.

The program to convert the data consists of three sections

(Figure 14). Five files are used:

Lab.dat - data input file,
Temp.dat - temporary lab data file,
His.dat - temporary his data file,
Index.dat - index of record and notebook number,
Reform.dat - final complete reformatted file.

In the first step, the input file is read and each record checked to determine if it is a lab or history record. If it is a lab record, the data is written to a direct access file Temp.dat in the order in which it will later be read. An index file, Index.dat, is created and the record number and lab notebook number are written to it. A history file, His.dat, is created and 'ZZZZZZZZ' is written to a record for each lab record. History records are skipped. Because history data has to be matched to the appropriate lab records, the lab data has to be written to a direct access file. But, because the format for the Datalib record consists of one field per line, the reformatted structure cannot be written to a direct access file. It would interpret each line as a record, so that line one is written in record one, line two is written in record two, etc. The lab data is written to Temp.dat without field tags.

The input file is then read again. Records are checked until a history record is located. When one is located, the index file, Index.dat, is read and checked until the notebook number for the corresponding lab record is located. The index tells in which record number the lab data is written. The history data is then written to that record number in His.dat, overwriting the Z's.

Finally, Temp.dat is read, one record at a time. The corresponding record number in His.dat is read. If the first eight characters in His.dat are Z's, the record from Temp.dat is written to a sequential file, Reform.dat, in the Datalib format, without history data. If the corresponding record in His.dat does not contain Z's, the record from Temp.dat and His.dat are written to Reform.dat in the same record in Datalib format, with history data. The program is set up so that field tags are not written for empty fields. The source program is shown in Appendix C. The program was run and Reform.dat created with the trial data.

Reform.dat, with trial data, was loaded to Datalib using Datalib's internal loading and indexing program. During the load, diagnostics were generated for each record loaded (Figure 15). If a particular field was not accepted, e.g., a date, the field name, problem and record identification were noted. The record in question was still loaded, but the field in question was not. By searching Datalib by the recid identified in the diagnostic log, that particular record can be retrieved and corrected if needed. The date fields in the current data will generate errors because dates were entered with zeros for months and/or years. Datalib cannot interpret zeros as a month or year. Any corrections needed will be done online after the load.

Once the data was loaded into Datalib, it was searched by various strategies to verify that all the data had been correctly loaded. The record type was first searched to verify the number of records loaded (Figure 16). Other searches and record changes were tried (Figures 17, 18). The data loaded correctly. The file was then further manipulated during the next two weeks, by the person who will be responsible for the file, to verify that the formats and functions met all those needed.

The full data file was transferred from the mainframe to the VAX by a batch listing. 13,170 records (lab plus his) were transferred. The data was run through Deltab.for to convert the tabs to spaces. On the first run through Convert.for, an error occurred which had not been anticipated. The program encountered a fatal error when two history records where found that did not have corresponding lab records. An additional file, Check.dat, was added to the program. If a history record was found that did not have lab records, the notebook number was written to Check.dat and the program continued. It was later discovered that the notebook numbers in the two history records that failed had been incorrectly input in the original data. The old system did not require that a history record have a corresponding lab record. It merely added a history record. Iteration numbers and record numbers were printed to the terminal, and in the last section of the program, to a file, to check the progress of the program and to verify the number of records processed. The index and check files were checked after the run to verify the number of records processed.

The records were then loaded to Datalib in an interactive session over a weekend, and the diagnostic log printed. In addition to the date fields that we knew would fail, there were several microfilm number fields that failed. There were a few records where the microfilm number had been entered as 'XXXXX.' This field requires a unique number, so only the first one that occurred was accepted. The errors on the log will be corrected as needed.

Datalib was searched by record type to insure that all 11,655 records were loaded. Searches were performed on notebook number and assignee at random and compared to the last printout from the mainframe run. No errors or discrepancies were encountered. All the records were loaded correctly.

As of April 1, 1987, the lab notebook file is in full interactive use on Datalib.

LAB NOTEBOOK TRACKING SYSTEM

LAB NOTEBOOK INVENTORY TABLE

Notebook Number	Name	Clock Number	Assign Date	Dept	Comp Date	Current Location	Microfilm Date	Microfilm Info	Witness	Change Date
000123	DOE,X	001234	0176	Allentown	1182	Smith, Q	1282	A11111A	Y	01/07/83
000234	DOE,Q	002345	0277	Allentown	1081	L-VAULT	1181	B22222B	Y	12/12/81
000345	JONES,Z	000123	0380	222-3333	1183	L-VAULT	1283	A00022B	Y	02/12/84
001234	SMITH,Q	003456	0480	333-4444	1283	JONES,X	0184	B33333C	Y	06/03/85
002345	DOE,Z	003467	0581	333-4444	1084	L-VAULT	1184	A03459C	Y	02/23/85

Sort by Notebook Number for All Notebooks
Figure 1

RETAKE

The Operator has Determined that the Previous Frame is Unacceptable and Has Refilmed the Page in the Next Frame.

of records loaded (Figure 16). Other searches and record changes were tried (Figures 17, 18). The data loaded correctly. The file was then further manipulated during the next two weeks, by the person who will be responsible for the file, to verify that the formats and functions met all those needed.

The full data file was transferred from the mainframe to the VAX by a batch listing. 13,170 records (lab plus his) were transferred. The data was run through Deltab.for to convert the tabs to spaces. On the first run through Convert.for, an error occurred which had not been anticipated. The program encountered a fatal error when two history records where found that did not have corresponding lab records. An additional file, Check.dat, was added to the program. If a history record was found that did not have lab records, the notebook number was written to Check.dat and the program continued. It was later discovered that the notebook numbers in the two history records that failed had been incorrectly input in the original data. The old system did not require that a history record have a corresponding lab record. It merely added a history record. Iteration numbers and record numbers were printed to the terminal, and in the last section of the program, to a file, to check the progress of the program and to verify the number of records processed. The index and check files were checked after the run to verify the number of records processed.

The records were then loaded to Datalib in an interactive session over a weekend, and the diagnostic log printed. In addition to the date fields that we knew would fail, there were several microfilm number fields that failed. There were a few records where the microfilm number had been entered as 'XXXXX.' This field requires a unique number, so only the first one that occurred was accepted. The errors on the log will be corrected as needed.

Datalib was searched by record type to insure that all 11,655 records were loaded. Searches were performed on notebook number and assignee at random and compared to the last printout from the mainframe run. No errors or discrepancies were encountered. All the records were loaded correctly.

As of April 1, 1987, the lab notebook file is in full interactive use on Datalib.

LAB NOTEBOOK TRACKING SYSTEM

LAB NOTEBOOK INVENTORY TABLE

Notebook Number	Name	Clock Number	Assign Date	Dept	Comp Date	Current Location	Microfilm Date	Microfilm Info	Witness	Change Date
000123	DOE,X	001234	0176	Allentown	1182	Smith, Q	1282	A11111A	Y	01/07/83
000234	DOE,Q	002345	0277	Allentown	1081	L-VAULT	1181	B22222B	Y	12/12/81
000345	JONES,Z	000123	0380	222-3333	1183	L-VAULT	1283	A00022B	Y	02/12/84
001234	SMITH,Q	003456	0480	333-4444	1283	JONES,X	0184	B33333C	Y	06/03/85
002345	DOE,Z	003467	0581	333-4444	1084	L-VAULT	1184	A03459C	Y	02/23/85

Sort by Notebook Number for All Notebooks
Figure 1

LAB NOTEBOOK LISTINGS BY CURRENT HOLDER OR LOCATION
NOT INCLUDING BOOKS IN E-VAULT, L-VAULT, OR R&D1
NOT INCLUDING BOOKS WITH A BLANK CURRENT LOCATION

CURRENT	BOOK	CLOCK	ASSIGN	NAME	DEPT	COMP
LOCATION	NUM.	NUM	DATE	ASSIGNED		DATE
DOE,Q	001234	123456	0181	DOE,Q	111-2222	0383
DOE,X	002345	234567	0282	DOE,X	222-3333	1284
DOE,Z	012345	012345	0383	JONES,Q	222-3333	0484
JONES,Q	000234	345678	0484	JONES,Q	333-4444	1085
JONES,Z	003456	001234	1183	DOE,Q	100-6666	1184
SMITH,Q	004567	023456	1083	DOE,X	333-5555	1285
SMITH,X	000123	034567	1082	SMITH,Q	444-7777	1283

Sort by Current Holder for All Active Books
Figure 2

LAB NOTEBOOK LISTING BY ASSIGNED NAME
NOT INCLUDING BOOKS IN E-VAULT, L-VAULT, OR R&D1

NAME ASSIGNED	BOOK NUM	CLOCK NUMBER	ASSIGN DATE	DEPT	CURRENT LOCATION	COMP DATE
DOE,Q Q	0001234	001234	1186	111-2222	DOE,Q Q	
DOE,Q Q	0002345	001234	1186	111-2222	DOE,Q Q	
DOE,X X	0000123	000234	1176	222-3333	DOE,X X	1177
JONES,Q Q	0023456	012345	1086	333-4444	SMITH,Z	1186
SMITH,Z Z	0003456	003455	1076	444-5555	DOE,X X	0280

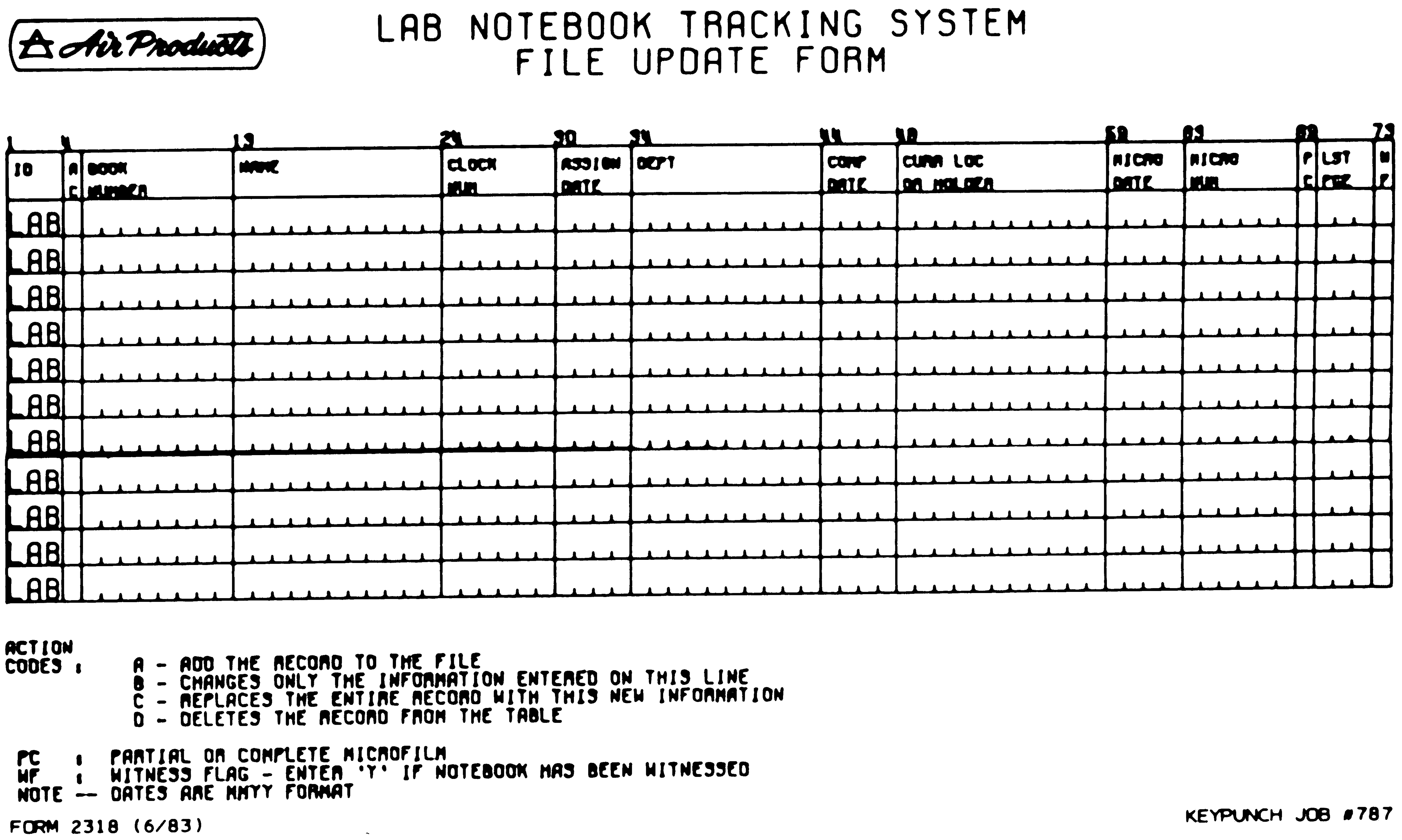
Sort by Original Assignee For All Active Books
Figure 3

LAB NOTEBOOK TRACKING SYSTEM

LAB NOTEBOOK HISTORY TABLE

Notebook	USER #1	RETURN DATE	USER #2	RETURN DATE	USER #3	RETURN DATE	USER #4	RETURN DATE	Change Date
001234	DOE,Z	0181	SMITH,Q	1282					12/23/82
002345	SMITH,Z	0284	JONES,Z	1185	DOE,Q				11/25/85
003456	JONES,Q								10/12/80
012345	SMITH,Q	1086	DOE,Q						11/01/86
023456	DOE,X	0284							02/26/84

Sort by Notebook Number For Reassignment History
Figure 4



Air Products

LAB NOTEBOOK TRACKING SYSTEM
FILE UPDATE FORM

1	4		13	24	30	34	44	48	58	63	69		73
ID	A C	BOOK NUMBER	NAME	CLOCK NUM	ASSIGN DATE	DEPT	COMP DATE	CURR LOC OR HOLDER	MICRO DATE	MICRO NUM	P C	LST PGE	W F
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													
LAB													

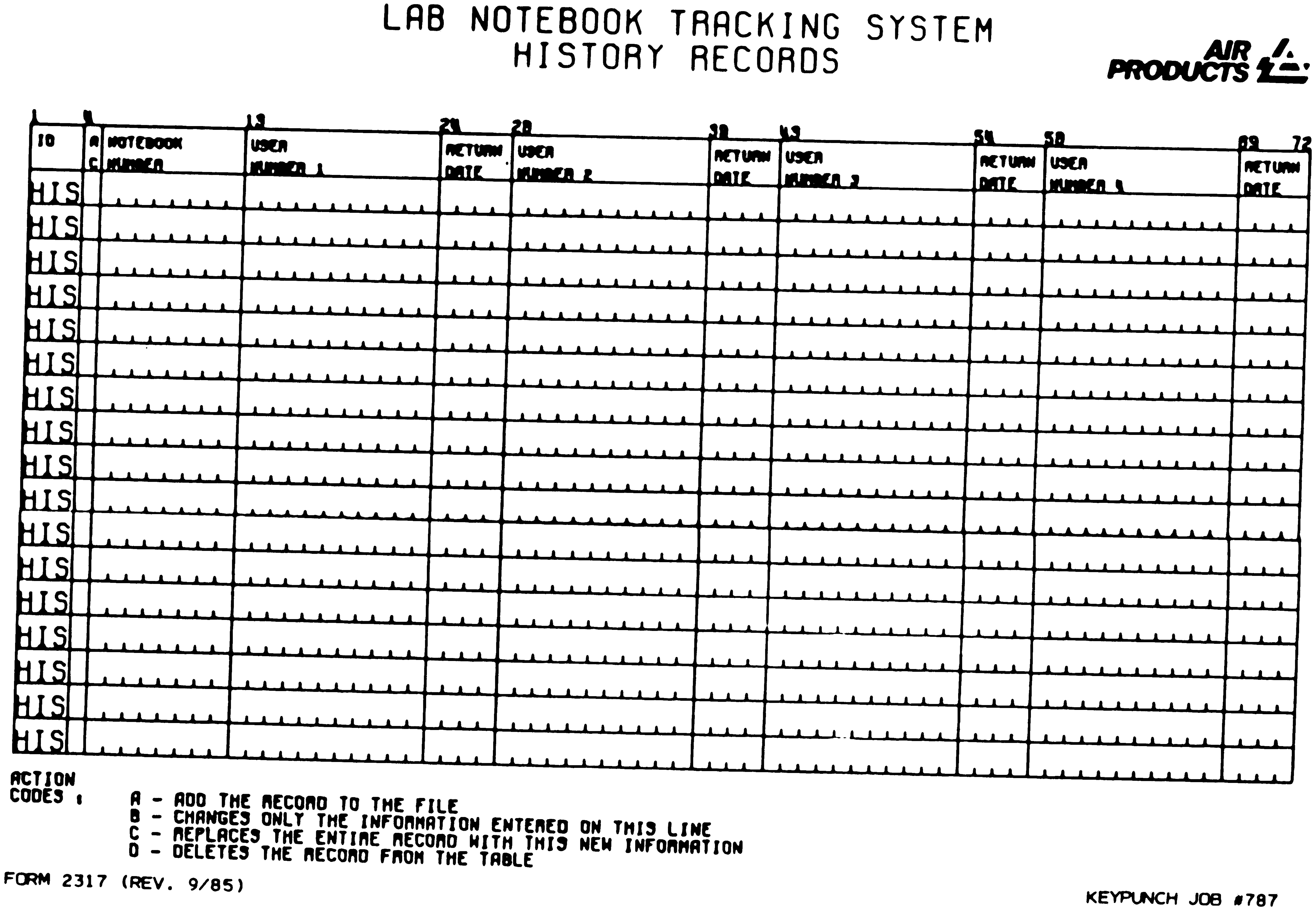
ACTION
CODES : A - ADD THE RECORD TO THE FILE
B - CHANGES ONLY THE INFORMATION ENTERED ON THIS LINE
C - REPLACES THE ENTIRE RECORD WITH THIS NEW INFORMATION
D - DELETES THE RECORD FROM THE TABLE

PC : PARTIAL OR COMPLETE MICROFILM
WF : WITNESS FLAG - ENTER 'Y' IF NOTEBOOK HAS BEEN WITNESSED
NOTE -- DATES ARE MMYY FORMAT

FORM 2318 (6/83)

KEYPUNCH JOB #787

Data Entry Update Form
Figure 5



LAB NOTEBOOK TRACKING SYSTEM
HISTORY RECORDS

AIR PRODUCTS

1	4		13	24	28	38	43	54	58	69 72
ID	A C	NOTEBOOK NUMBER	USER NUMBER 1	RETURN DATE	USER NUMBER 2	RETURN DATE	USER NUMBER 3	RETURN DATE	USER NUMBER 4	RETURN DATE
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										
HIS										

ACTION CODES :

A - ADD THE RECORD TO THE FILE
B - CHANGES ONLY THE INFORMATION ENTERED ON THIS LINE
C - REPLACES THE ENTIRE RECORD WITH THIS NEW INFORMATION
D - DELETES THE RECORD FROM THE TABLE

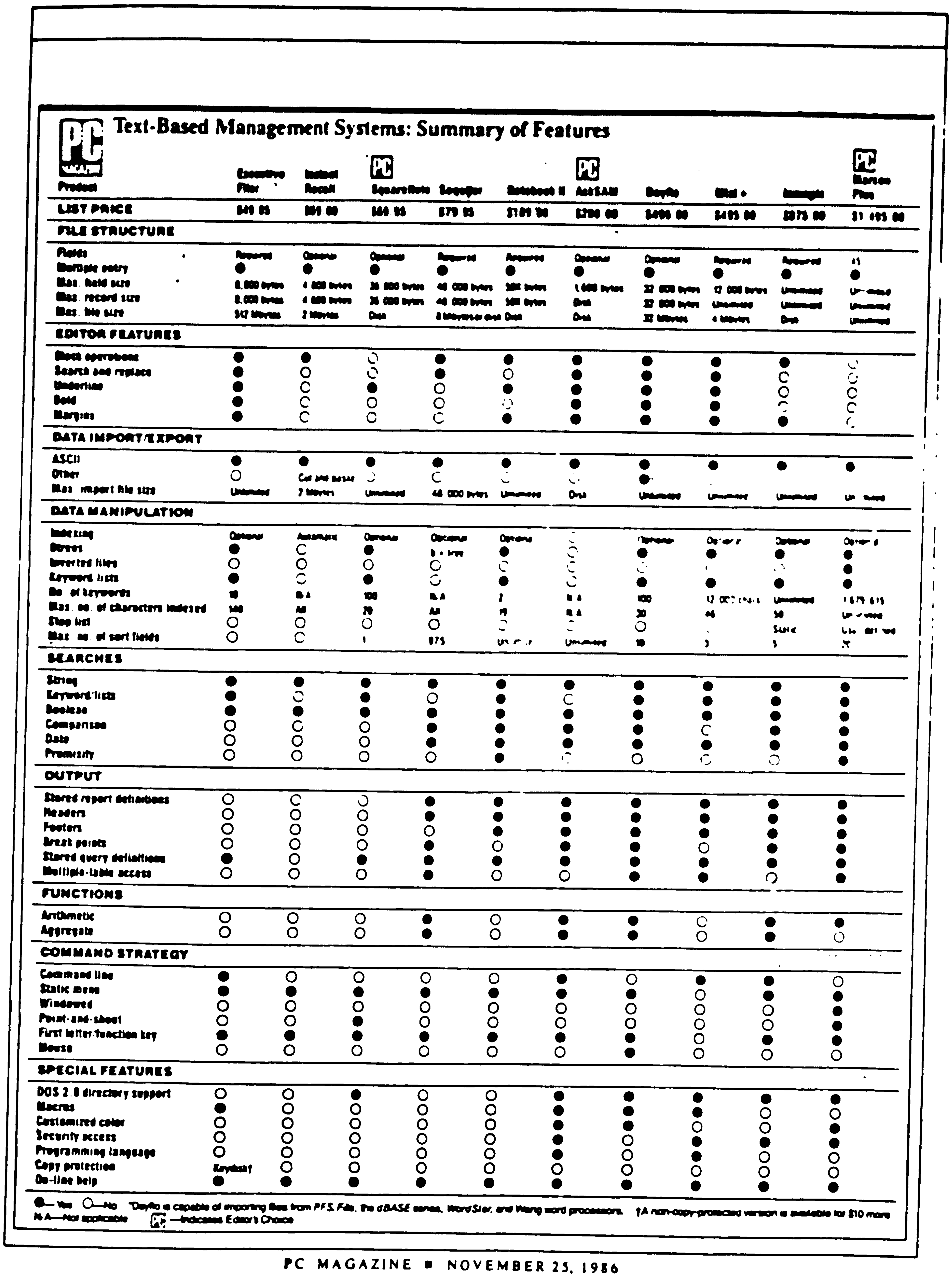
FORM 2317 (REV. 9/85)

KEYPUNCH JOB #787

Data Entry History Form
Figure 6

```
830706LABAAM1744P                          PISCATAWAY   SMITH,Q                                 A
861010LABA000197      DOE,Z            0172             L-VAULT                                 C
861010LABA000211      DOE,Z            1173             L-VAULT                                 C
861010LABA000212      DOE,Z            0275             L-VAULT                                 C
840510LABA001443      JONES,X                           L-VAULT                                 C
830615LABA001462      SMITH,X          0568         0671L-VAULT                                 A
830706LABA001744P                          PISCATAWAY   SMITH,Q                                 A
830627LABA001760      POTS,Q           1271         0475L-VAULT    0077A03138C                  A
830706LABBBT1744P                          PISCATAWAY   SMITH,Q                                 A
830706LABB001744P                          PISCATAWAY   SMITH,Q                                 A
830627LABB001760      POTS,Q           1271         0475L-VAULT    0077A03139C                  A
830706LABCCM1744P                          PISCATAWAY   SMITH,Q                                 A
830706LABC001744P                          PISCATAWAY   SMITH,Q                                 A
830706LABDDT1744P                          PISCATAWAY   SMITH,Q                                 A
830706LABD001744P                          PISCATAWAY   SMITH,Q                                 A
830706LABEE01744P                          PISCATAWAY   SMITH,Q                                 A
830706LABE001744P                          PISCATAWAY   SMITH,Q                                 A
830706LABFF01744P                          PISCATAWAY   SMITH,Q                                 A
830706LABF001744P                          PISCATAWAY   SMITH,Q                                 A
830718LABF004473L     JONES,X          0276LINWOOD  0779L-VAULT                                 A
830718LABF004474L     JONES,Z          0276LINWOOD  1277L-VAULT                                 A
830718LABF004475L     DOE,Z            0276LINWOOD  0779L-VAULT                                 A
830718LABF004476L     HOME,Q           0876LINWOOD  1277L-VAULT                                 A
830718LABF004478L     TOP,Z            1076LINWOOD  0580L-VAULT                                 A
```

Current File Structure
Figure 7



Text-Based Management Systems: Summary of Features

Product	Executive Filer	Instant Recall	SquareNote	Sequitur	Notebook II	AskSAM	Dayflo	[illegible] +	Inmagic	Marcon Plus
LIST PRICE	$49.95	$69.00	$69.95	$79.95	$189.00	$200.00	$495.00	$495.00	$975.00	$1,495.00
FILE STRUCTURE										
Fields	Required	Optional	Optional	Required	Required	Optional	Optional	Required	Required	45
Multiple entry	●	●	●	●	●	●	●	●	●	●
Max. field size	8,000 bytes	4,000 bytes	36,000 bytes	48,000 bytes	56K bytes	1,600 bytes	32,000 bytes	12,000 bytes	Unlimited	Unlimited
Max. record size	8,000 bytes	4,000 bytes	36,000 bytes	48,000 bytes	56K bytes	Disk	32,000 bytes	Unlimited	Unlimited	Unlimited
Max. file size	512 Kbytes	2 Mbytes	Disk	8 Mbytes or disk	Disk	Disk	32 Mbytes	4 Mbytes	Disk	Unlimited
EDITOR FEATURES										
Block operations	●	●	○	●	●	●	●	●	●	○
Search and replace	●	○	○	●	○	●	●	●	○	○
Underline	●	○	●	○	●	●	●	●	○	○
Bold	●	○	○	○	○	●	●	●	○	○
Margins	●	○	○	○	●	●	●	●	●	○
DATA IMPORT/EXPORT										
ASCII	●	●	●	●	●	●	●	●	●	●
Other	○	Cut and paste	○	○	○	○	●*			
Max. import file size	Unlimited	2 Mbytes	Unlimited	48,000 bytes	Unlimited	Disk	Unlimited	Unlimited	Unlimited	Unlimited
DATA MANIPULATION										
Indexing	Optional	Automatic	Optional	Optional	Optional	○	Optional	Optional	Optional	Optional
B-trees	●	○	●	B+ tree	●	○	●	●	●	●
Inverted files	○	○	○	○	○	○	○	○	○	●
Keyword lists	●	○	●	○	●	○	●	●	●	●
No. of keywords	10	N.A.	100	N.A.	2	N.A.	100	12,000 chars	Unlimited	1,679,615
Max. no. of characters indexed	140	All	20	All	19	N.A.	30	46	50	Unlimited
Stop list	○	○	○	○	○	○	○		Static	User defined
Max. no. of sort fields	○	○	1	975	Unlimited	Unlimited	10	3	5	[illegible]
SEARCHES										
String	●	●	●	●	●	●	●	●	●	●
Keyword lists	●	○	●	○	●	○	●	●	●	●
Boolean	●	●	●	●	●	●	●	●	●	●
Comparison	○	○	○	●	●	●	●	○	●	●
Date	○	○	○	●	●	●	●	●	●	●
Proximity	○	○	○	○	●	○	○	○	○	●
OUTPUT										
Stored report definitions	○	○	○	●	●	●	●	●	●	●
Headers	○	○	○	●	●	●	●	●	●	●
Footers	○	○	○	○	●	●	●	●	●	●
Break points	○	○	○	●	○	●	●	○	●	●
Stored query definitions	●	○	●	●	●	●	●	●	●	●
Multiple-table access	○	○	○	●	○	○	●	●	○	●
FUNCTIONS										
Arithmetic	○	○	○	●	○	●	●	○	●	●
Aggregate	○	○	○	●	○	●	●	○	●	○
COMMAND STRATEGY										
Command line	●	○	○	○	○	●	○	●	●	○
Static menu	●	●	●	●	●	●	●	○	●	●
Windowed	○	○	○	○	○	○	○	○	○	●
Point-and-shoot	○	○	●	○	○	○	○	○	○	●
First letter/function key	●	●	●	●	●	●	●	○	●	●
Mouse	○	○	○	○	○	○	●	○	○	○
SPECIAL FEATURES										
DOS 2.0 directory support	○	○	●	○	○	●	●	●	●	●
Macros	●	○	○	○	○	●	●	●	○	○
Customized color	○	○	○	○	○	●	●	○	○	●
Security access	○	○	○	○	○	●	○	●	●	●
Programming language	○	○	○	○	○	●	○	●	○	○
Copy protection	Keydisk†	○	○	○	○	○	○	○	○	○
On-line help	●	●	●	●	●	●	●	●	●	●

●—Yes ○—No *DayFlo is capable of importing files from *PFS:File*, the *dBASE* series, *WordStar*, and Wang word processors. †A non-copy-protected version is available for $10 more.
N.A.—Not applicable PC—Indicates Editor's Choice

PC MAGAZINE ■ NOVEMBER 25, 1986
213

Reprinted from PC Magazine
November 25, 1986

TBMS Features Summary
Figure 8

```
LIBLIST   LIST
   MIS
   RD1
   BUS    BUSINESS
   RD3
   ADM5
   IR     IRON RUN
   LAW
   ADM1-4
   MED    HEALTH U
   PURCH  CORP PURCH
   HOME   HOMETOWN
   PROC   PSG ENG
   STAND  STANDARDS
   RD2
   SAF    SAFETY
   TAX
   EMM    EMMAUS
   IGD    IGD ENG
   IA     INT AUDIT
   ARC    ARCAIR
   GARD   GARDNER
   GOV    GOVT SALES
   HOU    PSG HOUDRY
   IGDT   IGD TRAFFIC
   PAT    PATENT
YNLIST    LIST
   Y YES
   N NO
!ELEMENTS
LIBRARY CODED,KEYED;LIST=LIBLIST;SYN=$LIBCODE
LOAN_DATE DATE,KEYED
RETURN_DATE DATE, KEYED
CREATION DATE,KEYED;SYN = $CD
OPID ALPHA,KEYED;SYN = $OP
RECTYPE RECTYPE, KEYED; SYN = $RT
NOTES ALPHA, KEYED,TEXT
OWNER CODED,KEYED;LIST=LIBLIST;SYN=$OWNER; DEF = RD1
VOLID INTEGER,KEYED;SYN=$VOLID
RECID INTEGER,KEYED,UNIQUE;SYN = $ID
COPID INTEGER,KEYED;SYN = $COPID
TIMELAST DATE,KEYED;SYN = $TLM
LOCATION ALPHA,KEYED
```

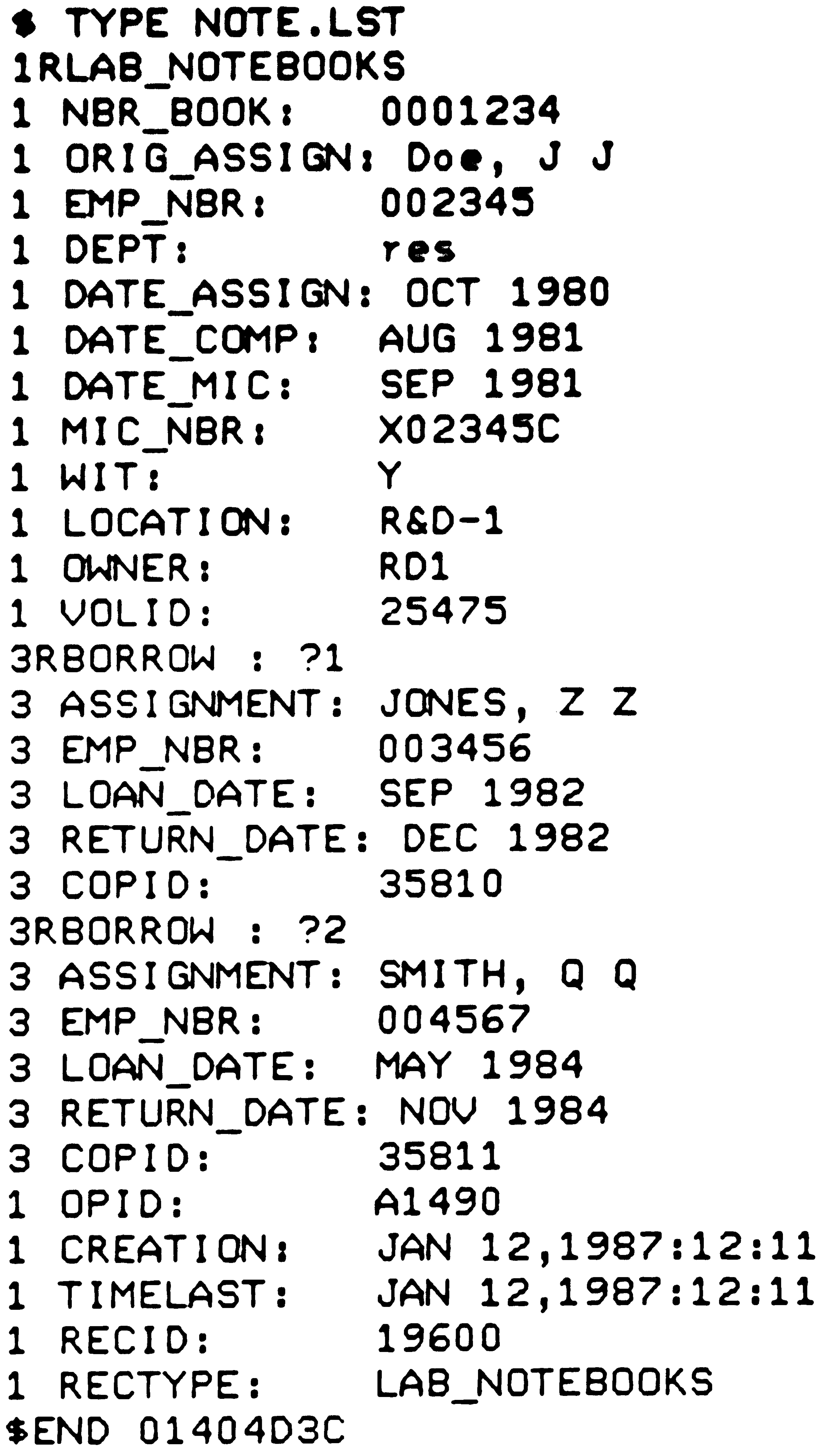
Datalib DDF
Figure 9

```
NBR_BOOK ALPHA,KEYED,UNI
ORIG_ASSIGN ALPHA,KEYED
DATE_ASSIGN DATE,KEYED
DATE_COMP DATE,KEYED
DATE_MIC DATE,KEYED
MIC_NBR ALPHA,KEYED,UNI
WIT_CODED,KEYED;LIST = YNLIST
EMP_NBR ALPHA,KEYED
ASSIGNMENT ALPHA,KEYED

!NESTED RECORDS

BORROW RECORD; CLASS=1;PROMPT = BORROWER DATA; FILE = NONE
   ASSIGNMENT
   EMP_NBR
   LOAN_DATE
   RETURN_DATE
   COPID
```

Datalib DDF
Figure 9

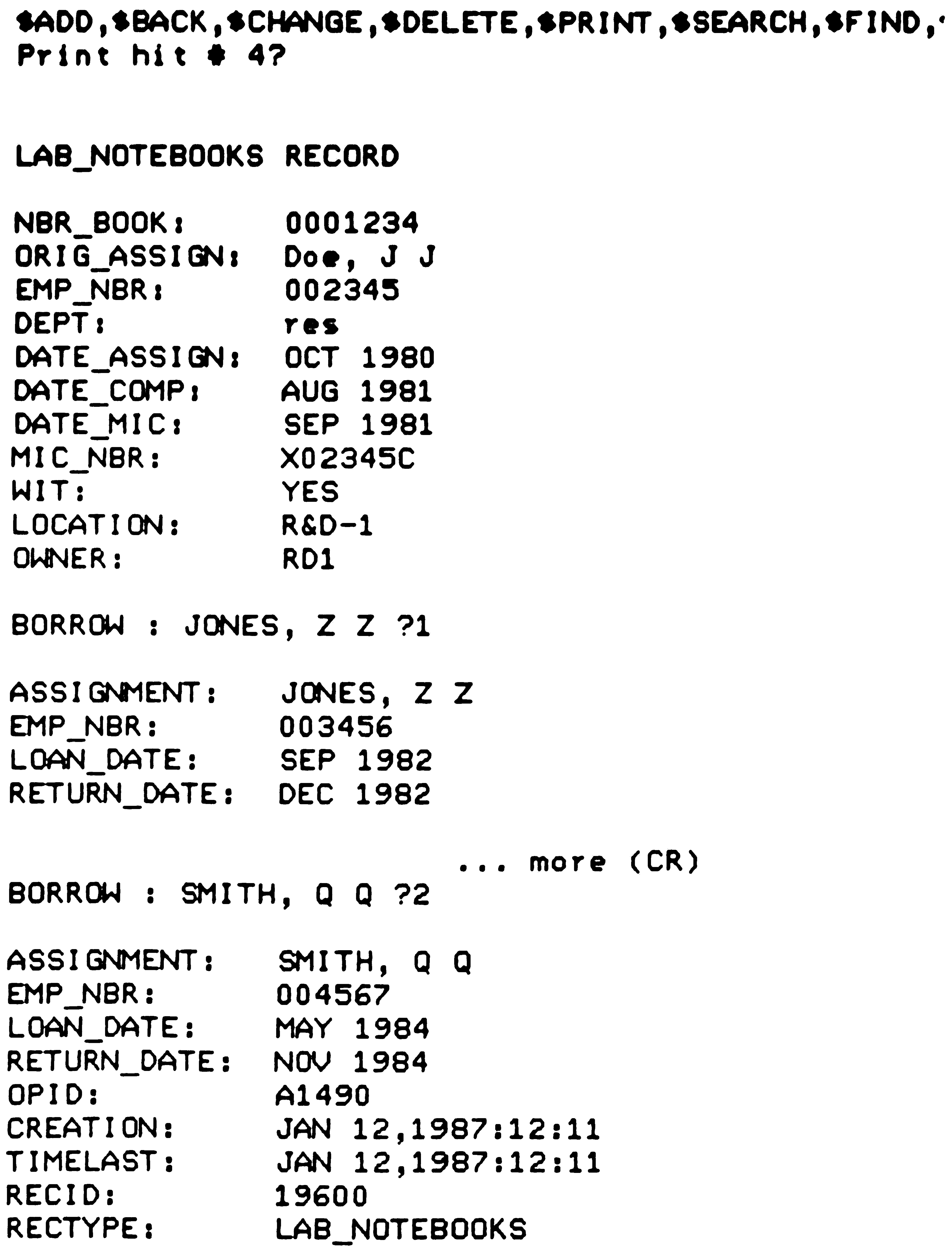
```
!RECORD TYPES

LAB_NOTEBOOKS RECORD;CLASS=1; FILE = BIBL; PROMPT = LAB_NOTEBOOKS
   NBR_BOOK, MAN
   ORIG_ASSIGN, MAN
   EMP_NBR
   DEPT
   DATE_ASSIGN
   DATE_COMP
   DATE_MIC
   MIC_NBR
   WIT
   LOCATION
   NOTES, REP
   LIBRARY
   OWNER
   VOLID
       BORROW, REP
   OPID;CREATION;TIMELAST;RECID;RECTYPE, MAN
```

Datalib Completed DDF Record
Figure 10



```
$ TYPE NOTE.LST
1RLAB_NOTEBOOKS
1 NBR_BOOK:    0001234
1 ORIG_ASSIGN: Doe, J J
1 EMP_NBR:     002345
1 DEPT:        res
1 DATE_ASSIGN: OCT 1980
1 DATE_COMP:   AUG 1981
1 DATE_MIC:    SEP 1981
1 MIC_NBR:     X02345C
1 WIT:         Y
1 LOCATION:    R&D-1
1 OWNER:       RD1
1 VOLID:       25475
3RBORROW : ?1
3 ASSIGNMENT:  JONES, Z Z
3 EMP_NBR:     003456
3 LOAN_DATE:   SEP 1982
3 RETURN_DATE: DEC 1982
3 COPID:       35810
3RBORROW : ?2
3 ASSIGNMENT:  SMITH, Q Q
3 EMP_NBR:     004567
3 LOAN_DATE:   MAY 1984
3 RETURN_DATE: NOV 1984
3 COPID:       35811
1 OPID:        A1490
1 CREATION:    JAN 12,1987:12:11
1 TIMELAST:    JAN 12,1987:12:11
1 RECID:       19600
1 RECTYPE:     LAB_NOTEBOOKS
$END 01404D3C
```

Datalib File Structure
Figure 11



```
$ADD,$BACK,$CHANGE,$DELETE,$PRINT,$SEARCH,$FIND,'
Print hit # 4?


LAB_NOTEBOOKS RECORD

NBR_BOOK:       0001234
ORIG_ASSIGN:    Doe, J J
EMP_NBR:        002345
DEPT:           res
DATE_ASSIGN:    OCT 1980
DATE_COMP:      AUG 1981
DATE_MIC:       SEP 1981
MIC_NBR:        X02345C
WIT:            YES
LOCATION:       R&D-1
OWNER:          RD1

BORROW : JONES, Z Z ?1

ASSIGNMENT:     JONES, Z Z
EMP_NBR:        003456
LOAN_DATE:      SEP 1982
RETURN_DATE:    DEC 1982

                           ... more (CR)
BORROW : SMITH, Q Q ?2

ASSIGNMENT:     SMITH, Q Q
EMP_NBR:        004567
LOAN_DATE:      MAY 1984
RETURN_DATE:    NOV 1984
OPID:           A1490
CREATION:       JAN 12,1987:12:11
TIMELAST:       JAN 12,1987:12:11
RECID:          19600
RECTYPE:        LAB_NOTEBOOKS
```

Datalib Online Record
Figure 12

Costs

File conversion - designing program, file conversion, testing 70 hrs. at $50/hr	$ 3,500
Personnel training	100
	$ 3,600

Benefits - over 5 years

Tangible

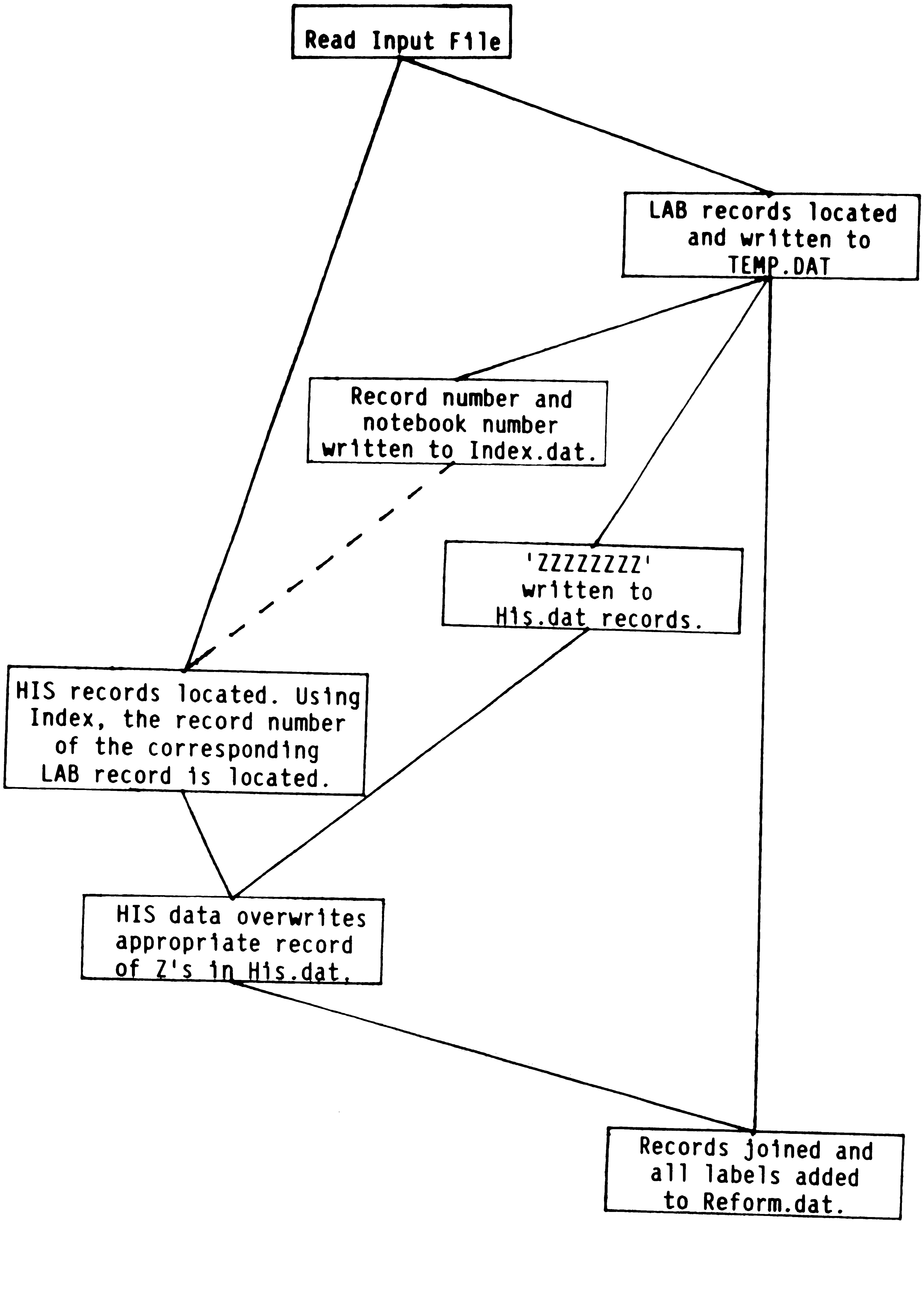
Elimination of manual operations 22.8 hours/yr at $15/hour	$ 1,710
Faster processing of operations 24 hours/yr at $15/hour	1,800
Reduced volume of paper produced and handled	990
Future cost avoidance $2,500/yr	12,500

Intangible

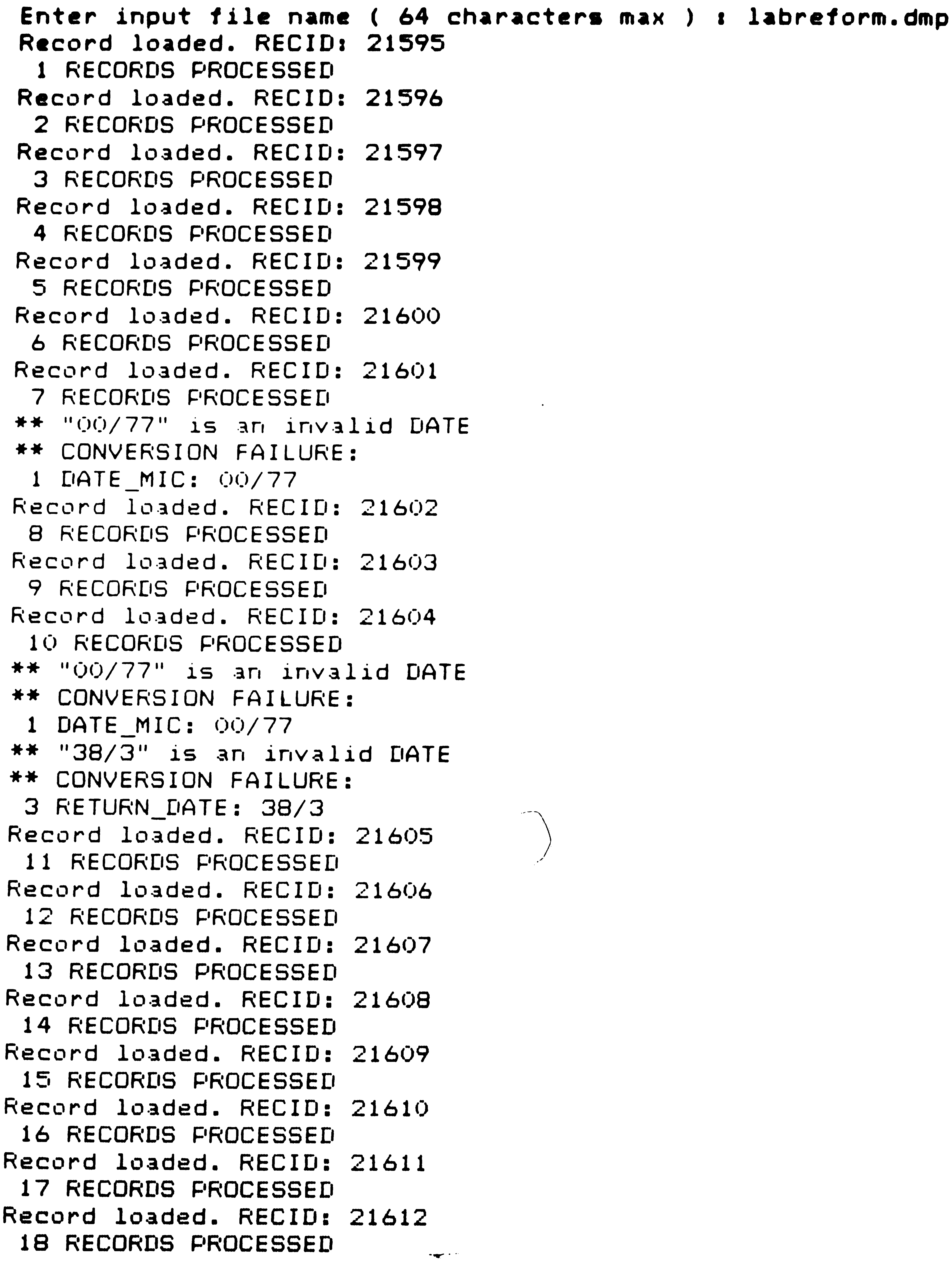
Smoothing operational flows - updating, retrieval
Rise in level of service quality and performance
Expansion capability
Compiling of statistics
More efficient use of personnel
Increased control of proprietary information

$17,000

Cost/Benefit Analysis Factors
Figure 13

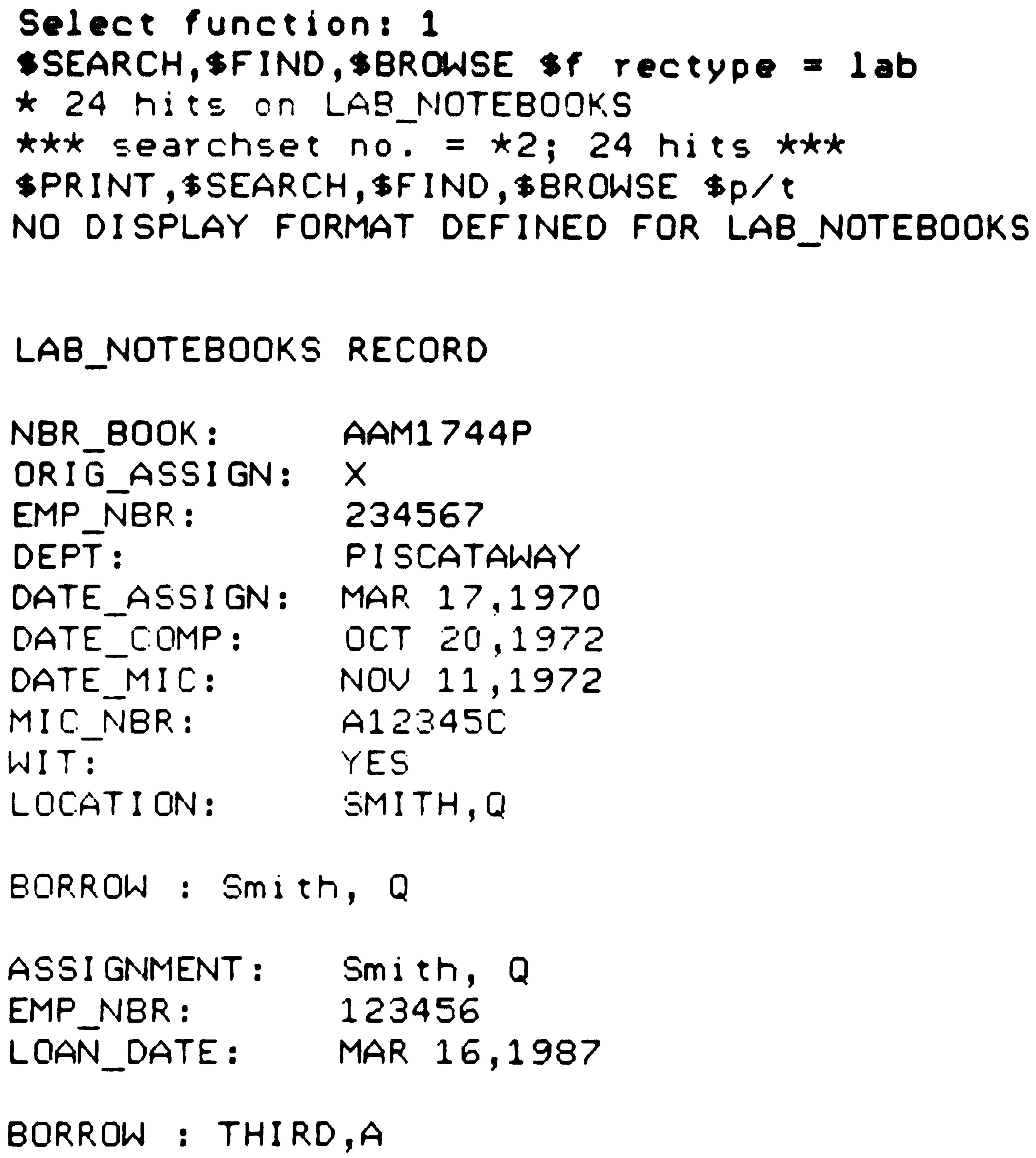

Read Input File
LAB records located and written to TEMP.DAT
Record number and notebook number written to Index.dat.
'ZZZZZZZZ' written to His.dat records.
HIS records located. Using Index, the record number of the corresponding LAB record is located.
HIS data overwrites appropriate record of Z's in His.dat.
Records joined and all labels added to Reform.dat.

Conversion Program Scheme
Figure 14



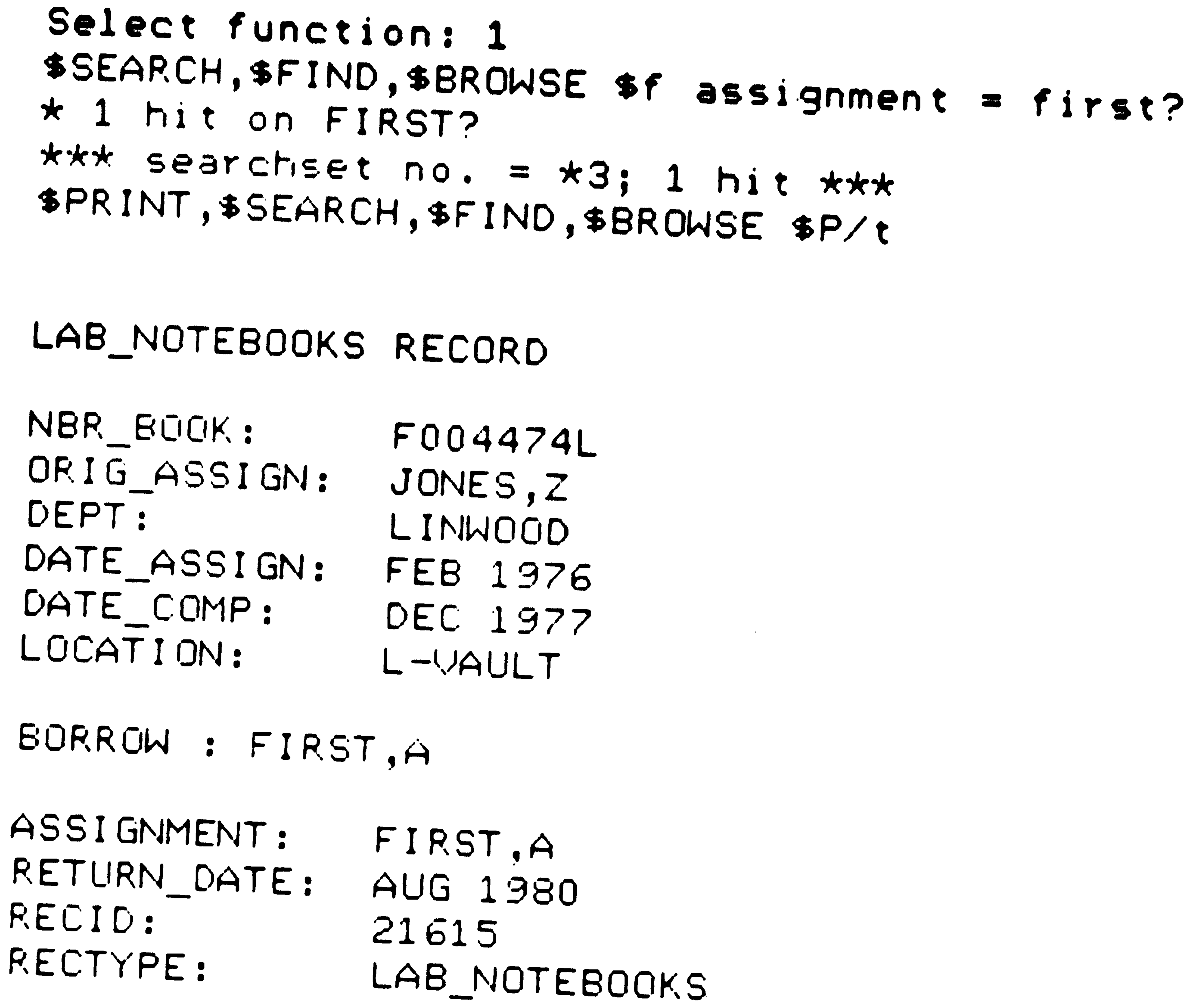
```
Enter input file name ( 64 characters max ) : labreform.dmp
Record loaded. RECID: 21595
 1 RECORDS PROCESSED
Record loaded. RECID: 21596
 2 RECORDS PROCESSED
Record loaded. RECID: 21597
 3 RECORDS PROCESSED
Record loaded. RECID: 21598
 4 RECORDS PROCESSED
Record loaded. RECID: 21599
 5 RECORDS PROCESSED
Record loaded. RECID: 21600
 6 RECORDS PROCESSED
Record loaded. RECID: 21601
 7 RECORDS PROCESSED
** "00/77" is an invalid DATE
** CONVERSION FAILURE:
 1 DATE_MIC: 00/77
Record loaded. RECID: 21602
 8 RECORDS PROCESSED
Record loaded. RECID: 21603
 9 RECORDS PROCESSED
Record loaded. RECID: 21604
 10 RECORDS PROCESSED
** "00/77" is an invalid DATE
** CONVERSION FAILURE:
 1 DATE_MIC: 00/77
** "38/3" is an invalid DATE
** CONVERSION FAILURE:
 3 RETURN_DATE: 38/3
Record loaded. RECID: 21605
 11 RECORDS PROCESSED
Record loaded. RECID: 21606
 12 RECORDS PROCESSED
Record loaded. RECID: 21607
 13 RECORDS PROCESSED
Record loaded. RECID: 21608
 14 RECORDS PROCESSED
Record loaded. RECID: 21609
 15 RECORDS PROCESSED
Record loaded. RECID: 21610
 16 RECORDS PROCESSED
Record loaded. RECID: 21611
 17 RECORDS PROCESSED
Record loaded. RECID: 21612
 18 RECORDS PROCESSED
```

Load Diagnostics
Figure 15



```
Select function: 1
$SEARCH,$FIND,$BROWSE $f rectype = lab
* 24 hits on LAB_NOTEBOOKS
*** searchset no. = *2; 24 hits ***
$PRINT,$SEARCH,$FIND,$BROWSE $p/t
NO DISPLAY FORMAT DEFINED FOR LAB_NOTEBOOKS


LAB_NOTEBOOKS RECORD

NBR_BOOK:       AAM1744P
ORIG_ASSIGN:    X
EMP_NBR:        234567
DEPT:           PISCATAWAY
DATE_ASSIGN:    MAR 17,1970
DATE_COMP:      OCT 20,1972
DATE_MIC:       NOV 11,1972
MIC_NBR:        A12345C
WIT:            YES
LOCATION:       SMITH,Q

BORROW : Smith, Q

ASSIGNMENT:     Smith, Q
EMP_NBR:        123456
LOAN_DATE:      MAR 16,1987

BORROW : THIRD,A
```

Datalib Search by Record Type
Figure 16



```
Select function: 1
$SEARCH,$FIND,$BROWSE $f assignment = first?
* 1 hit on FIRST?
*** searchset no. = *3; 1 hit ***
$PRINT,$SEARCH,$FIND,$BROWSE $P/t


LAB_NOTEBOOKS RECORD

NBR_BOOK:       F004474L
ORIG_ASSIGN:    JONES,Z
DEPT:           LINWOOD
DATE_ASSIGN:    FEB 1976
DATE_COMP:      DEC 1977
LOCATION:       L-VAULT

BORROW : FIRST,A

ASSIGNMENT:     FIRST,A
RETURN_DATE:    AUG 1980
RECID:          21615
RECTYPE:        LAB_NOTEBOOKS
```

Datalib Search
Figure 17

```
Select function: 1
$ADD,$BACK,$SEARCH,$FIND,$BROWSE $f nbr_book=f004474l
* 1 hit on F004474L
*** searchset no. = *2; 1 hit ***
$ADD,$BACK,$CHANGE,$DELETE,$PRINT,$SEARCH,$FIND,$BROWSE $add note
$ADD,$BACK,$CHANGE,$DELETE,$PRINT,$SEARCH,$FIND,$BROWSE $add note

LAB_NOTEBOOKS RECORD

NOTES : 1: This note is added as a test.
NOTES : 2: $done
$DONE,$ADD,$CHANGE,$DELETE,$PRINT,$ABANDON ** $done
Changes committed
$ADD,$BACK,$CHANGE,$DELETE,$PRINT,$SEARCH,$FIND,$BROWSE $p/t


LAB_NOTEBOOKS RECORD

NBR_BOOK:        F004474L
ORIG_ASSIGN:     JONES,Z
DEPT:            LINWOOD
DATE_ASSIGN:     FEB 1976
DATE_COMP:       DEC 1977
LOCATION:        L-VAULT
NOTES:           This note is added as a test.
OWNER:           RD1
VOLID:           27441

BORROW : FIRST,A ?1

ASSIGNMENT:      FIRST,A
RETURN_DATE:     AUG 1980
COPID:           37152
OPID:            A1683
CREATION:        MAR 9,1987:12:19
TIMELAST:        MAR 20,1987
RECID:           21615
```

Datalib Search & Record Change
Figure 18

Function/Software	DATALIB	TECHLIB	BIBLIOTECH
Cataloging	record fields defined by customer can handle records for a variety of materials new record and latest revisions are date stamped	record fields are defined by the customer new records and latest revision are date stamped modifications can be made to specified fields without reloading database can handle a variety of records (e.g., books, maps, AV material)	customer defines MARC tags to be transferred to BIBLIOTECH fields record flexible for a variety of materials
Retrieval	search fields defined by customer can browse authority file truncation keyword, boolean, truncation, words within a string, proximity range searching user can define print format	searchable fields defined by by customer keyword, boolean and format range searching display format of search results can be specified search history can be displayed	keyword, boolean, proximity,right and left truncation user can define print

Function/Software	DATALIB	TECHLIB	BIBLIOTECH
Retrieval (Cont'd)	search history displayable	search results can be sorted	
Authority Control	can be browsed by user global changes made through authority control includes cross references	thesaurus can be purchased separately; accommodates up to 13 relationships (broader terms, narrower terms, etc.)	authority file for staff use; broader, narrower and related terms; scope notes global changes made through authority control
Circulation	handles barcodes and manual updates patron directory displays books currently held by patrons temporary records can be entered for items like ILLS	handles barcodes and manual updates generates hold notices displays books currently charged to a patron	handles barcodes and manual updates patron directory generates hold notices statistical analysis of circulating materials can be done displays books currently held by patrons
Acquisitions	detailed fund accounting vendor directory	searching of acquisitions can be turned on or off provides vendor address file	generates purchase order record put in catalog

Function/Software	DATALIB	TECHLIB	BIBLIOTECH
Acquisitions (Cont'd)	copy records combined as a line item to vendor record put in catalog catalog		copy records to a given vendor and for a given title are gathered together as a line item
Serials	due for release in June 1985	in development	check-in routing union list records incorporated in catalog
Vendor Support	vendor installed and maintained via a maintenance agreement	80 hours of application support in purchase price maintenance agreement for updates	maintenance agreement available for updates

Function/Software	Ulisys	Atlas
Cataloging	maintains MARC format	maintains MARC records as separate file
	handles media types covered by MARC	provides direct transfer of OCLC records into ATLAS
	customer specific MARC fields to be transferred ULISYS fields	data stamps new records and latest revisions
	records can be stored in in MARC format or ULISYS format or both	customer defines MARC tags to be transferred to ATLAS fields.
Retrieval	truncation of beginning words in title, subject, author	four versions of catalog for searching, from menu to keywords and boolean searching
	keywords in author/title combination	searchable fields defined by customer
		sortable fields specified by customer
Authority Control	authority file contains "see", "see also," and and "see from" cross references	includes cross reference
		global change made through authority control

Function/Software	Ulisys	Atlas
Authority Control (Cont'd)	global changes made through authority control	
Circulation	handles barcodes and manual update	handles barcodes and manual update
	patron directory	provides patron directory
	temporary file can be created for ILLS	generates hold messages
	generates hold notices	displays books currently charged to a patron
Acquisitions	in development	detailed fund accounting
		acquisition record part of catalog
		record information transferred to and from catalog
		vendor directory provided
Serials	planned	union list of serials

Function/Software	Ulisys	Atlas
Vendor Support	vendor installed and maintained via a maintenance agreement	vendor installed and maintrained via a maintenance agreement
Technical Environment	All vendors update software to current VMS release TECHLIB runs on both VAX and Wang equipment	

```
C
C  PROGRAM TO ADD SPACES FOR THE TABS IN INPUT DATA
C
      CHARACTER *132 LINE, OBJ *1
C
      INTEGER I,J,K,LL,M,N,P,X
C
      OPEN (UNIT=1,FILE='TAB.DAT',STATUS='OLD',RECL=132,
     + ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGECONTROL='LIST'
      OPEN (UNIT=2,FILE='LAB.DAT',STATUS='NEW',RECL=132,
     + ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGECONTROL='LIST'
C
      DO 50 X=1,14000
      READ (1,FMT=10,END=1000) LINE
   10 FORMAT (A132)
      DO 40 I=1,100
        OBJ=LINE(I:I)
        IF (ICHAR(OBJ).NE.9) GO TO 40
        LINE(I:I)=' '
        J=I
          DO 15 K=1,12
            N=(K*8)+1
            IF (J.EQ.N) GO TO 20
            IF (J.LT.N) GO TO 25
   15     CONTINUE
   20   LL=7
        GO TO 30
   25   LL=N-J-1
   30     DO 35 M=100,J+1,-1
            P=M+LL
            LINE(P:P)=LINE(M:M)
   35     CONTINUE
        LINE(J+1:J+LL)='       '
   40 CONTINUE
      WRITE (2,45) LINE
   45 FORMAT (A132)
      WRITE (6,*) X
   50 CONTINUE
 1000 CLOSE (UNIT=1,DISP='SAVE')
      CLOSE (UNIT=2,DISP='SAVE')
      STOP
      END
```

Space Insertion Program
Appendix B

```
C
C   PROGRAM TO CONVERT MAINFRAME LABNOTEBOOK FILE TO DATALIB
C    BY D. ROSCHE' , FEB 1987
C
      CHARACTER *2 AD,ADD,CD,CDD,MD,MDD,DTA,DTAA,DTB,DTBB,DTC,
     + DTCC,DTD,DTDD,BN *8,NA *11,CN *6,DE *10,CL *11,MN *10,
     + WI *1,ID *3,USA *11,USB *11,USC *11,USD *11,HBN *8,CID *3,
     + CDT *6,D *8,E *8,NUL *11
C
      INTEGER X,Y,Z,A,B,C,R,RN
C
      OPEN (UNIT=1, FILE='LAB.DAT',STATUS='OLD',RECL=100,
     + ACCESS='SEQUENTIAL',MAXREC=15000)
C
      OPEN (UNIT=2,FILE='HIS.DAT',STATUS='NEW',RECL=101,
     + ACCESS='DIRECT',FORM='FORMATTED',CARRIAGECONTROL='LIST')
C
      OPEN (UNIT=3,FILE='TEMP.DAT',STATUS='NEW',RECL=100,CARRIAGE
     +CONTROL='LIST',ACCESS='DIRECT',FORM='FORMATTED')
C
      OPEN (UNIT=7,FILE='REFORM.DAT',STATUS='NEW',RECL=80,CARRIAGE
     +CONTROL='FORTRAN',ACCESS='SEQUENTIAL',FORM='FORMATTED')
C
      OPEN (UNIT=8,FILE='INDEX.DAT',STATUS='NEW',RECL=17,
     +ACCESS='DIRECT',FORM='FORMATTED',CARRIAGECONTROL='LIST')
C
      OPEN (UNIT=9,FILE='CHECK.DAT',STATUS='NEW',RECL=10,
     +ACCESS='SEQUENTIAL',FORM='FORMATTED',CARRIAGECONTROL='LIST')
C
C
C  RECORD READ FROM MASTER LIST, IF IT IS A HISTORY RECORD
C      IT IS SKIPPED.  RECORD IS REARRANGED AND WRITTEN TO
C      TEMP.DAT..  RECORD NO. AND NOTEBOOK NO. ARE WRITTEN
C      TO INDEX.DAT. RECORD NO. AND ZZZZZZZZ ARE WRITTEN TO
C      HIS.DAT.
C
      B=1
      DO 25 X=1,15000
        READ (1,FMT=5,END=1000) ID,BN,NA,CN,AD,ADD,DE,CD,CDD,
     +    CL,MD,MDD,MN,WI
    5   FORMAT (6X,A3,A8,6X,A11,A6,A2,A2,A10,A2,A2,A11,A2,A2,A10,A
        CID='HIS'
        IF (ID.EQ.CID) GO TO 25
C
      WRITE (3,REC=B,FMT=10)BN,NA,CN,DE,AD,ADD,CD,CDD,MD,MDD,MN,WI
   10 FORMAT (A8,A11,A6,A10,A2,A2,A2,A2,A2,A2,A10,A1,A11)
C
```

```
      WRITE (8,REC=B,FMT=15) B,BN
   15 FORMAT (I7,2X,A8)
C
C  Z'S ARE WRITTEN TO HIS.DAT
C
      D='ZZZZZZZZ'
      WRITE(2,REC=B,FMT=20) D
   20 FORMAT (A8)
C
      WRITE (6,*) 'X',X,'   B',B
C
      B=B+1
   25 CONTINUE
C
C  HISTORY DATA ,WHEN IT EXISTS, IS WRITTEN TO HIS.DAT, OVERWRITTI
C                            THE 'Z' RECORDS.
C
 1000 A=1
      REWIND 1
      DO 50 Y=1,15000
       READ (1,FMT=30,END=2000)CDT,ID,HBN,USA,DTA,DTAA,USB,DTB,DTB
     +   DTC,DTCC,USD,DTD,DTDD
   30  FORMAT (A6,A3,A8,6X,A11,A2,A2,A11,A2,A2,A11,A2,A2,A11,A2,A2
        CID='LAB'
        IF (ID.EQ.CID) GO TO 50
C
        R=1
   35   READ (8,REC=R,FMT=40,ERR=6000) RN,BN
   40   FORMAT (I7,2X,A8)
        IF (HBN.EQ.BN) GO TO 43
        R=R+1
        GO TO 35
C
   43   WRITE (2,REC=RN,FMT=45)HBN,USA,DTA,DTAA,USB,DTB,DTBB,USC,
     +   DTC,DTCC,USD,DTD,DTDD
   45   FORMAT (A8,A11,A2,A2,A11,A2,A2,A11,A2,A2,A11,A2,A2)
C
        WRITE (5,*) 'R',R,'  Y',Y
C
        A=A+1
        GO TO 50
 6000 WRITE (9,6001) HBN
 6001 FORMAT (A8)
C
      WRITE (6,*) 'R',R,'   Y',Y
C
   50 CONTINUE
```

```
C
C  LAB RECORDS ARE READ AND IT IS DETERMINED FROM THE CORRESPONDIN
C      HISTORY RECORD IF HISTORY DATA EXISTS.  RECORDS ARE WRITTEN
C                  ACCORDINGLY TO REFORM.DAT.
C
 2000 C=1
      B=B-1
      NUL='          '
      DO 95 Z=1,B
        READ (2,REC=C,FMT=55) D
   55   FORMAT (A8)
        E='ZZZZZZZZ'
        IF (D.NE.E) GO TO 70
C
      READ (3,REC=C,FMT=60)BN,NA,CN,DE,AD,ADD,CD,CDD,MD,MDD,MN,WI,
   60 FORMAT (A8,A11,A6,A10,A2,A2,A2,A2,A2,A2,A10,A1,A11)
      WRITE (6,*) C
      WRITE (9,61) C
   61 FORMAT (I7)
      WRITE (7,FMT=65) BN,NA
   65 FORMAT(' 1RLAB_NOTEBOOKS'/' 1 NBR_BOOK: ',A8/' 1 ORIG_ASSIGN
     +,A11)
C
      IF (CN.EQ.NUL) GO TO 4000
      WRITE (7,FMT=200) CN
  200 FORMAT(' 1 EMP_NBR: ',A6)
C
 4000 IF (DE.EQ.NUL) GO TO 4001
      WRITE (7,FMT=201) DE
  201 FORMAT(' 1 DEPT: ',A10)
C
 4001 IF (AD.EQ.NUL)GO TO 4002
      WRITE (7,FMT=202) AD,ADD
  202 FORMAT(' 1 DATE_ASSIGN: ',A2,'/',A2)
C
 4002 IF (CD.EQ.NUL) GO TO 4003
      WRITE (7,FMT=203) CD,CDD
  203 FORMAT(' 1 DATE_COMP: ',A2,'/',A2)
C
 4003 IF (MD.EQ.NUL) GO TO 4004
      WRITE (7,FMT=204) MD,MDD
  204 FORMAT(' 1 DATE_MIC: ',A2,'/',A2)
C
 4004 IF (MN.EQ.NUL) GO TO 4005
      WRITE (7,FMT=205) MN
  205 FORMAT(' 1 MIC_NBR: ',A10)
C
```

```
 4005 IF (WI.EQ.NUL) GO TO 4006
      WRITE (7,FMT=206) WI
  206 FORMAT(' 1 WIT: ',A1)
C
 4006 IF (CL.EQ.NUL) GO TO 4007
      WRITE (7,FMT=207) CL
  207 FORMAT(' 1 LOCATION: ',A11)
C
 4007 WRITE (7,FMT=208)
  208 FORMAT(' 1 OWNER: RD1'/' 1 RECTYPE: LAB_NOTEBOOKS'/' $END')
      GO TO 90
C
   70 READ (3,REC=C,FMT=75)BN,NA,CN,DE,AD,ADD,CD,CDD,MD,MDD,MN,WI,
   75 FORMAT (A8,A11,A6,A10,A2,A2,A2,A2,A2,A2,A10,A1,A11)
C
      READ (2,REC=C,FMT=80)HBN,USA,DTA,DTAA,USB,DTB,DTBB,USC,DTC,D
     + USD,DTD,DTDD
   80 FORMAT (A8,A11,A2,A2,A11,A2,A2,A11,A2,A2,A11,A2,A2)
      WRITE (6,*) C
      WRITE (9,81) C
   81 FORMAT (I7)
C
      WRITE (7,FMT=85) BN,NA
   85 FORMAT(' 1RLAB_NOTEBOOKS'/' 1 NBR_BOOK: ',A8/' 1 ORIG_ASSIGN
     +,A11)
C
      IF (CN.EQ.NUL) GO TO 4008
      WRITE (7,FMT=209) CN
  209 FORMAT(' 1 EMP_NBR: ',A6)
C
 4008 IF (DE.EQ.NUL) GO TO 4009
      WRITE (7,FMT=210) DE
  210 FORMAT(' 1 DEPT: ',A10)
C
 4009 IF (AD.EQ.NUL) GO TO 4010
      WRITE (7,FMT=211) AD,ADD
  211 FORMAT(' 1 DATE_ASSIGN: ',A2,'/',A2)
C
 4010 IF (CD.EQ.NUL) GO TO 4011
      WRITE (7,FMT=212) CD,CDD
  212 FORMAT(' 1 DATE_COMP: ',A2,'/',A2)
C
 4011 IF (MD.EQ.NUL) GO TO 4012
      WRITE (7,FMT=213) MD,MDD
  213 FORMAT(' 1 DATE_MIC: ',A2,'/',A2)
C
 4012 IF (MN.EQ.NUL) GO TO 4013
```

```
      WRITE (7,FMT=214) MN
  214 FORMAT(' 1 MIC_NBR: ',A10)
C
 4013 IF (WI.EQ.NUL) GO TO 4014
      WRITE (7,FMT=215) WI
  215 FORMAT(' 1 WIT: ',A1)
C
 4014 IF (CL.EQ.NUL) GO TO 4015
      WRITE (7,FMT=216) CL
  216 FORMAT(' 1 LOCATION: ',A11)
C
 4015 WRITE (7,FMT=217)
  217 FORMAT(' 1 OWNER: RD1')
C
      IF (USA.EQ.NUL) GO TO 4016
      WRITE (7,FMT=218) USA
  218 FORMAT (' 3RBORROW: ?1'/' 3 ASSIGNMENT: ',A11)
C
 4016 IF (DTA.EQ.NUL) GO TO 4017
      WRITE (7,FMT=219) DTA,DTAA
  219 FORMAT (' 3 RETURN_DATE: ',A2,'/',A2)
C
 4017 IF (USB.EQ.NUL) GO TO 4018
      WRITE (7,FMT=220) USB
  220 FORMAT (' 3RBORROW: ?2'/' 3 ASSIGNMENT: ',A11)
C
 4018 IF (DTB.EQ.NUL) GO TO 4019
      WRITE (7,FMT=221) DTB,DTBB
  221 FORMAT (' 3 RETURN_DATE: ',A2,'/',A2)
C
 4019 IF (USC.EQ.NUL) GO TO 4020
      WRITE (7,FMT=222) USC
  222 FORMAT (' 3RBORROW: ?3'/' 3 ASSIGNMENT: ',A11)
C
 4020 IF (DTC.EQ.NUL) GO TO 4021
      WRITE (7,FMT=223) DTC,DTCC
  223 FORMAT (' 3 RETURN_DATE: ',A2,'/',A2)
C
 4021 IF (USD.EQ.NUL) GO TO 4022
      WRITE (7,FMT=224) USD
  224 FORMAT (' 3RBORROW: ?4'/' 3 ASSIGNMENT: ',A11)
C
 4022 IF (DTD.EQ.NUL) GO TO 4023
      WRITE (7,FMT=225) DTD,DTDD
  225 FORMAT (' 3 RETURN_DATE: ',A2,'/',A2)
C
 4023 WRITE (7,FMT=226)
```

```
  226 FORMAT (' 1 RECTYPE: LAB_NOTEBOOKS'/' $END')
C
   90 C=C+1
   95 CONTINUE
C
C  CLOSE ALL OPEN STATEMENTS
C
 3000 CLOSE (UNIT=1,DISP='SAVE')
      CLOSE (UNIT=2,DISP='SAVE')
      CLOSE (UNIT=3,DISP='SAVE')
      CLOSE (UNIT=7,DISP='SAVE')
      CLOSE (UNIT=8,DISP='SAVE')
      STOP
      END
```

Bibliography

Atre, S. Data Base: Structured Techniques for Design, Performance and Management, New York, 1980.

Bell, C. Margaret. "Laboratory Notebook Storage and Retrieval Systems", Science and Technology Libraries, V1, N4, (Summer 1981), 65-71.

Berg, John L. "Computer Science and Technology: Data Base Directions - The Conversion Problem", Proceeding of the Workshop of the National Bureau of Standards and the Association for Computing Machinery, November 1977, (September 1980).

Birss, Edward E. and James P. Fry. "Generalized Software for Translating Data", American Federation of Information Processing, Proceedings of the National Computer Conference, NY, V45, (June 1976), 889-897.

Birss, Edward W. "Database Conversion: A Position Paper", Database Conversion Technology Panel, (August 1977), 463-475.

Cartledge, Jr W. J. "Program and File Conversion", Datamation, V12, N6, (June 1966), 22-27.

Casanova, Marco A. and Jose E. Amaral de Sa. "Mapping Uninterpreted Schemes into Entity - Relationship Diagrams: Two Applications to Conceptual Schema Design", IBM Journal of Research and Development, V28, N1, (January 1984), 82-94.

Cibbanelli, Pamela and Carol Tenopir. "Labnotes A Laboratory Notebook Retrieval System", The Information Age in Perspective, Proceedings of the 41ST Annual Meeting, NY, V15, (November 1978), 67-69.

Datalib product summary, Data Sigma Corporation.

Datalib Users Manual - Draft, Version 9, April 1985.

FitzGerald, Jerry, Andra F. FitzGerald and Warren D. Stallings, Jr. Fundamentals of Systems Analysis, New York, Second Ed., 1981.

Grosch, Audry N. Minicomputers in Libraries, 1981-82: the Era of Distributed Systems, NY, 1982.

Hayes, L. Susan and Judith A. Vogel. "IMP-32 - An Interactive Management Program", Database, V5, N3, (August 1982), 56-62.

Jacobs, Barry E. Applied Database Logic, Volume 1 Fundamental Database Issues, New Jersey, 1985.

Murrill, Donald D. "Microfilming and Encoding Laboratory Notebooks at the Philip Morris Research Center", American Documentation Institute, Proceeding of the 1966 Annual Meeting, V3, 51-56.

Niemi, Timo. "Formal Restructuring Functions for Hierarchial Data Bases", International Journal of Computer and Information Sciences, V12, N6, (October 1983), 385-411.

Ott, Retha Z. "Developing a New System for Control of Laboratory Notebook Information - The Information Manager's Perspective", American Society for Information Science, Management of Information Systems Proceedings, (May 1978), 6.

Personal communication to Allergan Pharmaceuticals.

Personal communication to Howard Hoffman, RCA.

Puglia, Vincent. "TBMS Database Power Unleashed", PC Magazine, V5, N20, (November 1986), 211-230.

Ramirez, Jesus A. "Automatic Generation of Data Conversion - Programs Using a Data Description Language (DDL) Vol. 1 System Descriptions and Documentation", Moore School of Electrical Engineering, Philadelphia, 1973.

Ramirez, Jesus A. "Automatic Generation of Data Conversion - Programs Using a Data Description Language (DDL) Vol. 2 Appendices - User Guide", Moore School of Electrical Engineering, Philadelphia, 1973.

Ramirez, Jesus A., N.A. Rin, M. Brown and N. S. Prywes. "Automatic Generation of Data Conversion Programs Using a Data Description Language", Moore School of Electrical Engineering, Philadelphia, 1974.

Rosenberry, Phoebe and Barbara G. Prewitt. "Applications of System 2000 Data Base Management Software at Rohm & Haas Company Research Libraries", The Information Age In Perspective, Proceedings of the 41ST ASIS Annual Meeting, NY, V15, (November 1978), 274-277.

Seddon, John G. "Data Conversion Without Tears", Datamation, V30, N5, (April 1984), 130-135.

Biographical Sketch

Deborah was born in Abington, Pennsylvania, to Edgar and Dorothy Yoder in November, 1958. She attended Shippensburg University from 1976 to 1980, and graduated with a B.A. in Chemistry. Deborah has been working for Air Products and Chemicals, Inc. since 1980 and has held the postion of Information Specialist since 1982. She started studies at Lehigh University part time in the fall of 1981. She is married and is currently living in Emmaus.