MARC ON THE CDC 6400: A TEACHING TOOL
FOR LIBRARY CLASSIFICATION

by

Carolyn Ann Kacena

A Thesis Submitted to the Faculty of the
DEPARTMENT OF COMPUTER SCIENCE
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
In the Graduate College
THE UNIVERSITY OF ARIZONA

1974
STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Carolyn Kaena

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Robert L. Baker
Associate Professor of Systems and Industrial Engineering

April 29, 1974
ACKNOWLEDGMENTS

No thesis is ever done by one person and I would like to take the opportunity to partially discharge my indebtedness to the many others who helped and encouraged me along the way in completing this project.

My first thanks go to my Committee for their guidance and help: Dr. Robert L. Baker, chairman; Dr. Ralph E. Griswold and Dr. Margaret F. Maxwell.

Special acknowledgment must be given to Mr. Charles M. Peters, Assistant Librarian for Systems and Facilities, for making the funds for research available in support of the program development. For time and patience above and beyond the call of duty, thanks also go to Mr. Hans R. Bart, Chief Catalog Librarian, and my other colleagues in the Catalog Department for allowing me to test out the MARCA system on them and to tell them more than they ever wanted to know about MARC!

The Consultants' Office in the University Computer Center was also most helpful. Of the many pleasant assistants in the UCC special thanks must be given to Mr. Richard A. Holmes and Mr. Donald R. Brandt for use of special subroutines they had written which saved months of project time.
Further listings of the individuals who helped would fill reams of paper. Miss Anne Lincoln, who had the job of painstakingly typing the manuscript, including sample computer entries in complete accuracy, a special thank you.

For my sister, Barbara, who played computer to check out the program with me, another, this time public, thank you. And on and on for the many family members, friends and colleagues who said just the right word at just the right time to keep me going: thank you.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>List of Illustrations</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>viii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
</tbody>
</table>

## CHAPTER

1. Introduction
   - Scope of the Work 2

2. MARC File Structure
   - History and Development 4
   - MARC Record Structure 8
     - Leader 8
     - Record Directory 13
     - Control Fields 13
     - Variable Fields 14
   - MARC Tape Structure 14
   - MARC Files Available 15
   - MARC at LC 16

3. MARC Utilization 19
   - Information Dynamics Corporation 20
   - Information Design 21
   - Ohio College Library Center 22
   - Oklahoma Department of Libraries--MARC-0 25
   - Indiana University Student Project 31
   - Other Projects 33

4. MARC CAPABILITIES 36
   - MARCTRS 36
   - MARCSET 39
   - MARCIND 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. MARCA IN L.S. 394</td>
<td>50</td>
</tr>
<tr>
<td>Class Assignment</td>
<td>50</td>
</tr>
<tr>
<td>Search by LC Class</td>
<td>51</td>
</tr>
<tr>
<td>Search by Range of LC Classes</td>
<td>51</td>
</tr>
<tr>
<td>Search by LC Subclass</td>
<td>52</td>
</tr>
<tr>
<td>Search by Range of LC Subclasses</td>
<td>52</td>
</tr>
<tr>
<td>Search by Individual Classification</td>
<td>53</td>
</tr>
<tr>
<td>Search by Range of Individual Numbers</td>
<td>54</td>
</tr>
<tr>
<td>6. CONCLUSIONS AND RECOMMENDATIONS</td>
<td>55</td>
</tr>
<tr>
<td>APPENDIX A: MARC VARIABLE FIELD TAGS</td>
<td>60</td>
</tr>
<tr>
<td>APPENDIX B: MARCTRS SOURCE PROGRAM</td>
<td>64</td>
</tr>
<tr>
<td>APPENDIX C: MARCSET SOURCE PROGRAM</td>
<td>74</td>
</tr>
<tr>
<td>APPENDIX D: MARCIND SOURCE PROGRAM</td>
<td>96</td>
</tr>
<tr>
<td>APPENDIX E: ACRONYMS USED IN THIS THESIS</td>
<td>132</td>
</tr>
<tr>
<td>SELECTED BIBLIOGRAPHY</td>
<td>135</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Sample Library of Congress Card in the MARC II Format for 9-Track Tapes</td>
<td>9</td>
</tr>
<tr>
<td>2.</td>
<td>Card from Which Sample MARC Record Was Made</td>
<td>10</td>
</tr>
<tr>
<td>3.</td>
<td>Outline of Leader</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>MARC-0 S. D. I. Entry Sample</td>
<td>29</td>
</tr>
<tr>
<td>5.</td>
<td>Abbreviated MARC Record Sample</td>
<td>48</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MARC Project Participants</td>
<td>7</td>
</tr>
<tr>
<td>2. Fixed-Length Data Elements</td>
<td>43</td>
</tr>
</tbody>
</table>
ABSTRACT

This thesis presents the development of a series of computer programs for instructional use in Library Science, known as MARCA. MARCA is a batch-process, selective-dissemination-of-information system which provides broad subject access to the Library of Congress Machine Readable Cataloging (MARC). The student presents the operator with a search strategy based on the Library of Congress classification and receives responses in an abbreviated MARC format. The output is then used in Advanced Cataloging class to study the relationships of Dewey and Library of Congress classification and subject heading analysis.

The thesis contains a survey of the history and development of the MARC Distribution Service, uses which the MARC tapes have been put to, a description of the MARCA system and a discussion on the results which have obtained in the use of MARCA with the Advanced Cataloging class during the spring, 1974 semester. A listing of the variable fields which form the MARC record, listings of the source programs for the MARCA program segments and a list of acronyms used are also included.
CHAPTER 1

INTRODUCTION

The purpose of this thesis is to report the development and use of a series of computer programs which provide access to the MAchine Readable Cataloging (MARC) data issued by the Library of Congress (LC).

The series of programs developed at The University of Arizona, hereafter designated MARCA, is designed to serve as a batch-process, selective-dissemination-of-information (SDI) program based on the LC classification applied to the records on the tapes for students in the University of Arizona Graduate School of Library Science. The system and its related class assignments are used to aid the student in studying the MARC II format and the LC classification for books. After an introductory presentation of the MARC II format, the student formulates a search strategy in terms of an LC classification number. The fields selected from the records for print out are then used to study the relationship of the two library classifications, Dewey Decimal and Library of Congress, to each other and also to the subject headings used for the titles selected. There is no attempt made to print out the entire contents of a library card in the standard card format.
The predominant use to date of the MARC service has been to print catalog cards for library catalogs on demand. Thus, for most systems one must require the cataloging information for a particular title in order to retrieve information from the tapes. Access is usually by the LC card number or main entry. The MARCA programs attempt to identify materials on the file by subject content descriptor, LC classification number, in order to provide an alerting system for new titles in that subject area. The search profiles are a considerably simplified version of the Oklahoma SDI service and the class projects assigned in relation to the programs are similar to those of the Syracuse University School of Library Science during the MARC Pilot Project in 1968-69 and implemented on their IBM 360/50.

MARCA is written in the FORTRAN EXTENDED language with COMPASS subroutines and has been implemented on the Control Data Corporation (CDC) 6400, SCOPE Operating System 3.4.

Scope of the Work

This thesis consists of six chapters: Introduction, MARC File Structure, MARC Utilization, MARCA Capabilities, MARCA in L. S. 394, and Conclusions and Recommendations. In addition to the six chapters, there are five appendices: MARC Variable Field Tags, MARCTRS Program Listing, MARCSET
Program Listing, MARCIND Program Listing, and List of Abbreviations Used in the Thesis.

Chapter 2 is an overview of the file structure developed by the MARC Project at LC with a brief history of the project and of current MARC use at LC. Chapter 3 is a review of various uses of the MARC data base at other installations. Chapter 4 describes what each of the programs in the MARCA series is designed to accomplish. It also includes a description of the input requirements of each stage of the system. Chapter 5 summarizes the results of the program use with the Library Science 394 students during the spring semester 1974.

Appendix A lists the possible tags for variable field information in the MARC II format. Appendices B-D are listings of the source programs for the three stages of the project. Appendix E is the listing of acronyms and abbreviations used within the thesis.
CHAPTER 2

MARC FILE STRUCTURE

The MARC format, designed as a communications format, was the basis of the American National Standards Institute format structure for bibliographic information interchanges on magnetic tape, which has been recommended to the International Standards Organization for adoption.

History and Development

In 1963 the Library of Congress received the report, Automation in the Library of Congress, which recommended essentially that the Library investigate methods of automating the entire flow of its internal processing (U. S. Library of Congress 1963). A major component of such automation would be the development of a machine-readable file of the cataloging information produced at the Library. Inforonics, Inc. was asked to study and report on the means available to record such bibliographic data and published the report, The Recording of Library of Congress Bibliographic Data in Machine Form (Buckland 1965).

In January, 1965, the First Conference on Machine-Readable Catalog Copy was called to discuss the Inforonics
This meeting gave rise to two main conclusions (Mathies and Watson 1973, p. 135):

1) machine-readable catalog data from LC was essential to library automation efforts, and 2) a standard format was needed to allow other libraries to catalog their own works in a compatible mode.

Henriette Avram, Ruth Freitag and Kay Guiles of the LC staff were asked to develop a possible format for the machine data. In June, 1965, their analysis of cataloging data from a machine processing viewpoint was issued to the library world (Mathies and Watson 1973; Avram, Freitag and Guiles 1965). It discussed contents of the records, fixed and variable length fields, and how these concepts relate to the bibliographic data to be handled and how the data itself is represented internally in the computer. November 22, 1965, a meeting was held at LC for staff and representatives of interested libraries and library associations to discuss the proposed format.

LC had already asked the Council on Library Resources for funding of a project to explore the possibility of distribution of cataloging data in machine-readable form; after this Second Conference on Machine-Readable Catalog Copy, further funding was elicited for development of a pilot project which involved the development of machine-readable data and distribution to a select group of libraries (Mathies and Watson 1973, pp. 134-135).
The Third Conference on Machine-Readable Catalog Copy was held in February, 1966 (MARC Development Office 1972g) and involved LC staff and representatives of the 16 libraries chosen as part of the pilot group (Avram 1968a, pp. 4-5). For a complete list, see Table 1. This group of libraries covers a broad spectrum of type of library (special, governmental, public, school, and university), from Washington, D. C., to Seattle, Washington, and points between. Each participant had to have personnel and funding available to take advantage on short notice of the machine-readable information (many were already automated in some portion of their library's functions), and they had to be willing to evaluate the tapes and prepare reports on their utilization. The February meeting was to establish project parameters and coordinate final suggestions on the format of the data.

Formal distribution of machine-readable bibliographic data began in October, 1966, with weekly service inaugurated in November of the same year. The tape contents were limited to English-language monographs currently cataloged by the LC, and by May, 1968, the MARC tapes had included over 44,000 titles (Avram 1968a, p. 7).

Evaluations of the MARC I format were initiated early in 1967 and the revised, MARC II, format was ready for presentation and discussion at the June, 1967 American
Table 1. MARC Project Participants

Includes libraries chosen in January, 1968 to participate until termination of the Pilot Project and designated with an asterisk.

<table>
<thead>
<tr>
<th>Library Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argonne National Laboratory</td>
</tr>
<tr>
<td>Georgia Institute of Technology</td>
</tr>
<tr>
<td>Harvard University</td>
</tr>
<tr>
<td>Indiana University</td>
</tr>
<tr>
<td>Montgomery County Public Schools, Rockville, Md.</td>
</tr>
<tr>
<td>Nassau (County) Library System, Hempstead, L. I.</td>
</tr>
<tr>
<td>National Agricultural Library</td>
</tr>
<tr>
<td>Redstone Scientific Information Center</td>
</tr>
<tr>
<td>Rice University</td>
</tr>
<tr>
<td>University of California Institute of Library Research</td>
</tr>
<tr>
<td>University of Chicago</td>
</tr>
<tr>
<td>University of Florida</td>
</tr>
<tr>
<td>University of Missouri</td>
</tr>
<tr>
<td>University of Toronto</td>
</tr>
<tr>
<td>Washington State Library</td>
</tr>
<tr>
<td>Yale University</td>
</tr>
<tr>
<td>*California State Library</td>
</tr>
<tr>
<td>*Cornell University Library</td>
</tr>
<tr>
<td>*Illinois State Library</td>
</tr>
<tr>
<td>*SUNY Biomedical Communications Network</td>
</tr>
</tbody>
</table>
Library Association (ALA) meeting. From October, 1967 until July, 1968, the MARC I format was under continuous revision and the tapes were released in an interim format. With the July, 1968 tapes, the MARC II format was used and is the form currently being used, with minor changes. It is undergoing modification primarily in the implementation of additional fields and subfield designators which were identified in early plans but not utilized immediately.

**MARC Record Structure**

The MARC II communications format record has four types of fields: Leader, Record Directory, Control, and Variable. See Figures 1 and 2 for a sample MARC record and its Library of Congress card.

**Leader**

The Leader is a fixed-length field of twenty-four characters. The first five characters designate the number of characters in the logical record. A logical record, which is the entire data contained on an LC printed card plus the fixed data elements, leader and directory information, is made up of one or more physical records of 2048 characters, a character limitation set by the machine on which the tapes are created. There is one leader per logical record, one record directory per logical record, and each physical record includes data up to the 2048th
<table>
<thead>
<tr>
<th>Leader</th>
<th>Record Directory</th>
</tr>
</thead>
</table>

| 00441 | n | a | m | 2 | 2 | 00133 | 864500 | 001001300000 | 008004100013 | 050001600054 |
| 082001300070 | 1000035000083 | 245008900118 | 260004800207 | 300001900255 | 6500034000274 |

Control Fields

<table>
<thead>
<tr>
<th>LC Card Number</th>
<th>Fixed Length Data Elements</th>
</tr>
</thead>
</table>

| 64 | 008443 | 681001 | s | 1964 | nyu | 8818 | $a | 897 | Walter | $d | 1897 | $e | 6ed. |

Variable Fields

<table>
<thead>
<tr>
<th>LC Call Number</th>
<th>Dewey Number</th>
<th>Main Entry</th>
</tr>
</thead>
</table>

| O | $a | PS614 | $b | L75 | F | $a | 811 | 5082 | F | 10 | $a | Lowenfels, Walter | $d | 1897 | $e | 6ed. |

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
</table>

| L | $a | Poets of today; $b | New American Anthology. $c | With a preface and an essay by B | Imprint |

| $a | Langston Hughes. F | 0 | $a | New York | $b | International Publishers | $c | 1964 |

<table>
<thead>
<tr>
<th>Collation</th>
<th>Subject Heading</th>
</tr>
</thead>
</table>

| 843 | $p | c21 | $c | cm. | F | 80 | $a | American Poetry | $y | 20th Century |

\( \text{\textbackslash n} \) = blank \( \text{\textbackslash f} \) = field terminator \( \text{\textbackslash r} \) = end of record

Figure 1. Sample Library of Congress Card in the MARC II Format for 9-Track Tapes
Lowenfels, Walter, 1897- ed.
143 p. 21 cm.

1. American poetry—20th cent.  i. Title.

PS614.L75 811.5082 64–8443
Library of Congress

Figure 2. Card from Which Sample MARC Record Was Made
character (MARC Development Office 1972a, pp. 3-6). The character count is followed by a one-character record status designator which is alphabetic, i.e., n for new record, p for previously made CIP entry, etc. This is followed by the designator for type of record (a=language material, printed) and a designator for bibliographic level (m=monographic) and two blanks. These are followed by the indicator count (2 for monographs) and subfield count (2 for monographs) which alert the programmer to the number of characters which start each variable field and the number of characters used as subfield delimiters within the variable field (for seven-track tapes the subfield delimiter requires three characters since the delimiter itself is in the nonstandard character set I and requires the shift character to introduce it). The base address of the data (five characters) follows, with the next character the encoding level indicator (either 6 for creation from physical inspection of the title, 1 for LC catalog entry, or 8 for CIP information). After two more blanks, the entry map section gives a 4 for length of field data in the directory record, 5 for the length of starting address data in the directory record, and two zeros which are currently undefined. Figure 3 shows the Leader layout.
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>16</th>
<th>17</th>
<th>20</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Length</td>
<td>Status</td>
<td>Type of Record</td>
<td>Bibliographic Level</td>
<td>Blanks</td>
<td>Indicator Count</td>
<td>Subfield Code Count</td>
<td>Base Address of Data</td>
<td>Encoding Level</td>
<td>Blanks</td>
<td>Entry Map</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Outline of Leader
Record Directory

This is a series of 12-character entries which identify for the user which variable fields are to be found within the logical record (three-character tag); how many characters are in the particular field, including the field terminator (four characters); and the starting address of the data field in the record (five characters). The record directory section ends with a field terminator (MARC Development Office 1972a, pp. 6-7).

Control Fields

The control fields contain alpha-numeric data, usually of fixed length, with a directory entry and tag but without the indicator or subfield coding of the variable fields (MARC Development Office 1972a, pp. 8-22). The tag is allowable only once in a logical record. The data fields implemented to date are fields 001 and 008; the control field 002—subrecord map—has not yet been used but is designated to handle supplements and/or special issues which are "dashed-on" the LC cards as bibliographically related items. Both fields 001 and 008 are printed in the output from MARCIND and their contents are described fully in Chapter 4.
Variable Fields

The variable fields are also alphanemic in character and terminate in a field terminator code (MARC Development Office 1972a, pp. 23-57). They may or may not have more than one subfield delimiter and may (in some cases) be repeated when necessary in a logical record. They are always introduced by a two-character indicator which may be a pair of blanks or may have significant characters to be incorporated in processing the data in the field. MARCSET retrieves and re-formats for use by MARCIND the variable fields 043, 050, 082, all 100 level tags, 245, 260, and 600-651. These are fully described in Chapter 4. Appendix A contains a listing of all the variable field tags provided by the MARC II format.

MARC Tape Structure

The 7-track tapes used in this project were obtained from the Library of Congress and are written in odd parity at 556 bpi. Each logical file is terminated by a tape mark (17'). The file organization includes a volume header label, a file header label, tape mark, file of logical records for monographs, tape mark, end-of-file label, and two tape marks. MARCA programs do not process either the introductory or the end-of-file labels. The tapes are created on an IBM 360/40 DOS (disk operating system). The extended library character set is provided for as either an
8-bit (9-channel tape) or a 6-bit (7-channel tape) series of codes. There are three major character sets of 64 characters each described with a fourth partial set for Greek, subscripts and superscripts. The following description of the 6-bit code is taken from the MARC Manuals Used by the Library of Congress, 2d ed. (Information Systems Office 1970c, v. 1, p. 4):

This code is derived by removing the 6th and 8th bit from the 8-bit code set. The standard 6-bit set includes lower case alphas, numerals, and most punctuation marks and other special characters. Three characters in this standard set are designated as non-locking shift codes of which only two (73g and 75g) will be used in the immediate future. The presence of one of these codes indicates that the next character is in either one of two non-standard 6-bit code sets. Code 73 precedes characters in non-standard set I. Code 75g precedes characters in non-standard set II.

The separate character sets for Greek letters, subscripts and superscripts are introduced and terminated by locking character sequences (e.g., 73g33g47g to access the Greek characters). These are further discussed in Chapter 4.

MARC Files Available

Individual formats have been developed and published within the MARC system for books, films, maps, and serials. Formats for manuscripts, music and sound recordings are being developed. Not all of these formats will become part of the distribution service from the LC but are being published to facilitate the communications of
information between other institutions by providing stand-
ardized specifications for data elements. Presently,
tapes are distributed for monographs (English and French),
serials, films, and maps ("MARC Distribution Service to
Expand" 1973, pp. 139-140).

The subscription data for fiscal year 1973-74 is
as follows ("MARC Distribution Service to Expand" 1973, pp.
139-140): total distribution service, $2,300; monographs
(all), $1,300; monographs (English), $1,000; films, $400;
maps, $400; and serials, $400. Annual files of English
monographs are available at $1,000 each, for 1968-date.
Films for 1972-73 are available for $400. Approximately
2,000 monographic records per week, 600 film records per
month, 350 map records per month, and 800 serial records
per month are available under the subscriptions. The
first tapes for French monographs and films were available

MARC at LC

The MARC format, while of continuing concern to
the library world in general, was from LC's point of view
a necessary cog in its own automation procedures; the tape
distribution service, as the card distribution services
earlier, is a by-product of LC's efforts to process
materials for its own collections. The Card Service has
maintained stockpiles of cards produced by the Library since the sale of cards began in 1903.

The Library's Multiple Use MARC System (MUMS), which is the total system encompassing MARC at LC, had its initial applications in on-line correction procedures of the MARC records which were created under the format recognition procedures developed as part of the RECON (REtrospective CONversion) project. In addition, MUMS provides on-line input and access to the Process Information File, which identifies items which the Library has received for cataloging but which are still in the catalog backlog. Authority file systems currently proposed will be linked with the MARC bibliographic record in such a way that book catalogs will automatically print the necessary references for headings used on titles in the collection. Plans are also underway internally to prepare book catalogs for various LC reading rooms. The first catalog proposed for printing on the Government Printing Office linotrot from MARC tapes is the Library of Congress Catalog: Motion Pictures and Filmstrips (Avram, Maruyama and Rather 1972, p. 235).

The Card Automated Reproduction and Distribution System (CARDS) has been implemented fully through Phase I. This is the automated order handling and inventory control stage whereby Card Division subscribers are each assigned
an OCR (optical character recognition) identification number which is preprinted on card order blanks. LC card number is to be printed in OCR format on the order blanks and these special forms facilitate the handling of orders by the Card Service. Data generated from these orders assist the Division in maintaining its card stocks based on user demands.

Phase II of CARDS is implemented on a limited basis: this involves the use of the VideoComp to produce card sets for records on the MARC data base on demand rather than stockpiling cards for potential orders. If the entire MARC data base were currently implemented through the Phase II procedures, approximately 40 percent of the card orders placed to LC would be handled on an on-demand basis (Avram et al. 1972, p. 287). This statement, made while MARC was still limited to English language monographs cataloged after 1968, gives great promise to university libraries as the French monographs and other Roman alphabet languages are implemented.

In total, the RECON and MARC Distribution Services combined have made over 426,525 titles available in machine-readable form by January, 1974 ("MARC Development Office" 1974, p. 11).
CHAPTER 3

MARC UTILIZATION

The MARC format, often in a modified form for local processing, has been used in many libraries and processing centers around the country and around the world since it was first developed, either in the Library of Congress MARC II format directly or in one of the national modifications, e.g., Canadian, British, French, Japanese, etc. (Chauveinc 1970, 1971; Coward 1969; MARC II Specifications 1969).

The MARC User Survey, 1972 (MARC Development Office 1972h) showed 54 MARC subscribers as of August 8, 1972, in the United States and Canada. Many of these subscribers provide access to the data base to their own subscribers. For example, 12 MARC subscribers provide printed access to selected MARC records on demand to a total of 177 agencies, both internal and external; magnetic tape output from six subscribers is used by 51 agencies; and three subscribers provide on-line access to MARC on demand to 58 agencies (MARC Development Office 1972h, p. 35). Below are descriptions of selected MARC users, either current or part of the MARC Pilot Project, which show how the materials have been
manipulated and, in particular, how access for library
school students and/or SDI services have been implemented.

Information Dynamics Corporation

Barely operational is the Information Dynamics
Corporation (IDC) BIBNET on-line bibliographic search and
cataloging service. The local installation connection is
a mini-computer terminal, hardcopy printer, and telephone.
The user can enter the MARC data base via LC card number,
ISBN (International Standard Book Number), first letters
of main entry and title (apparently similar to the key
developed by the Ohio College Library Center [OCLC] and
discussed fully in that section), or a key word from the
title of the work. If the full record is available on the
MARC base, the cathode ray screen will display the title
and main entry for verification that it is the record
desired. If it is, the full MARC record can be displayed
and then transferred to a cassette in the local terminal
for later editing and printing of card sets. If the full
record is not available, but the LC card number is avail-
able in IDC's microfiche catalog of all National Union
Catalog entries from 1953-date, the user can request a set
of cards be made at the central office in Reading, Mass.
Once a card set has been made for one patron, if it is in
Roman characters, the entry is transcribed into the
complete MARC file for subsequent users to access on-line. The Technical Applications and News Announcements for January, 1974, quotes the data base as having 430,000 MARC records from LC, including Cataloging in Publication (CIP) entries, plus card numbers for all 3,335,000 catalog records produced by LC since 1953. Telephone connection to the data base is required for the search segment of processing; the editing and printing function can be handled on the local mini-computer after all searches have been completed ("BIBNET Goes On-Line" 1974; "BIBNET's Instant Cataloging Service" 1974).

Information Design

Another commercial user of the MARC tapes for card production is Information Design, Menlo Park, California. This company uses the MARC tapes to create microfilm images of the cards, complete with overtyped headings, which print out six cards per sheet. These can be searched via a microfilm index of card numbers and title entries, the appropriate frame focused on the Xerox microprinter, and printed six cards per sheet. The cards are then torn apart and alphabetized, ready for filing. The University of Arizona Catalog Department has studied this system with a two-month test period of indexes alone and predicts a 36% match to our current cataloging work load (Kacena, Phipps and Tepper 1973). The cost of the system is such
that work load coverage must be in the range of 60% before serious thought of implementation will be given. Increased implementation on MARC of foreign language cataloging will make this and similar services more attractive locally.

Ohio College Library Center

The Ohio College Library Center (OCLC) is a not-for-profit corporation chartered by Ohio in 1967 to provide access to the MARC data by member libraries and union catalog capabilities for improvement of the interlibrary loan system within the state. It is the first library consortium developed specifically to utilize the MARC tapes in catalog card production. Originally limited to college libraries in Ohio, terminal access is now spread across the country either directly to the Columbus, Ohio computer or to other regional centers providing access to similar files.

Terminal owners can access the MARC data base and the shared cataloging base of other OCLC users through a centralized Xerox Sigma 5 computer. The user can search the file by LC card number, by title search, or by author-title search key. The title search key is the first three letters of the first non-article in the title, followed by the first letter of the following three words. For example, the title Influence of Librarians in Liberal Arts Colleges
in Selected Decision-making Areas would have the key: inf,0,1,i. All of the commas must be present even if the title is shorter than four words. The author-title search key is created from the first three characters of the main entry and the first non-English-article word of the title. For example: Grace, William J. Response to Literature would have the search key: gra,res (Lyons and Northcraft 1972, p. 266).

Studies by Frederick G. Kilgour and his staff on author-title keys showed that the 3,3 search key provided five or fewer hits 99.08% of the time when tested against a file of 132,808 entries (Kilgour, Long and Leiderman 1970). When further tested against the MARC file, the cumulative probability for up to five entries per key was 98.91% (Long and Kilgour 1971).

Similarly, title search keys of 3,3 yield eight or fewer entries 99% of the time from a file of 16,792 title-only entries (Kilgour et al. 1971). Kilgour contends that the title-only searches are on first order Markov strings which yield more matches due to correlations of the two consecutive words in a title which do not occur in author-title searches. Experiments with 3,1,1; 3,1,1,1; 3,1,1,1,1 yielded 12, five and four hits per search at the 99% range. Since specificity did not improve significantly with the addition of a fifth word, their final choice for
search from title only was the 3,1,1,1 (Long and Kilgour 1972, pp. 18-19).

When a hit is made in the file, the total MARC record is visible on the CRT screen. The system includes a field for library identification marks which shows every library in the OCLC system which has cataloged the particular title. The user can modify the MARC record as necessary for local conditions but the record which future users see is the unmodified LC record.

Cards can be called for once the record is modified for local use. They are printed in Columbus during the night shift and shipped, arranged in filing order, the next morning. For example, if The University of Arizona were a user of the system, the screen would show the MARC information for a title being added to the Humanities Department about Thomas Wolfe. The record would be matched to the copy of the work in the Catalog Department and any information which needed change would be modified at that time. For example, if the call number created a conflict in our shelf list, the necessary change would be affected prior to requesting cards be printed. Depression of the print key on the terminal causes the OCLC computer to print the modified record for The University of Arizona, Humanities Department. Parameters determined when the Library joined OCLC would be already programmed such that the cards
for our main catalog would arrive in three sorted files: shelflist cards in classification number order; author-title entries interfiled alphabetically; and subject cards interfiled in a third alphabetical sequence. Cards would also have been produced for the Humanities Department catalog, but they would arrive in two files: shelflist in call number order and the remainder of the card set (subjects as well as author[s] and title[s]) interfiled alphabetically since they have a dictionary, not a divided catalog. Thus, card format and filing sequence are chosen by the member library as they join to fit previously developed arrangements.

The OCLC system has been replicated in New England by NELINET (New England Library Network) and planned replications or direct access to Ohio is found in Texas, Georgia and New York. OCLC was planned as a regional node in a totally coordinated national system and has tried to maintain standard formats and data communications formats to facilitate coordination with other such regional endeavors.

Oklahoma Department of Libraries--MARC-0

MARC-0 is the acronym for the Oklahoma project begun in February, 1969, with the initiation of LC's Distribution Service. As of October, 1972, the MARC-0 data base was
operating on three 3330 disk packs with a total number of 235,440 records ("MARC-Oklahoma Data Base Storage and Retrieval Project Report, no. 8" 1972, p. 2). Beginning in October, 1971, the Southwestern Library Interstate Cooperative Endeavor (SLICE) has worked with MARC-0 to make the MARC-0 services known and utilized by libraries within the region of the Southwestern Library Association (SWLA).

Four major services are available through the SLICE/MARC-0 project (Oklahoma Department of Libraries and SLICE Office Director 1972a). The primary service is the "Cataloging Data Search and Print" segment. The search key is the LC card number for the particular title and the catalog data on the MARC file is printed in modified card format on paper stock in requested sequence and mailed to the requester. The service is processed twice weekly, in 1972, and the results are in the requester's hands approximately five working days after receipt of the search request in Oklahoma City. As of 1972, requests could be submitted on a printed form for keypunching or on pre-punched cards. Charges included a 10¢ charge for each keypunched request done in Oklahoma City, 30¢ for each printed citation. The minimum charge per batch was $5.00. When CIP information is available for the title, this
information is printed out as the response to the request unless it has been superceded by a complete card record.

The second service is the "Record Search and Copy" segment. Again the search is by LC card number and the output is a computer tape accompanied by the printed Reference Listing. The user then develops his own support programs for creating card sets, book catalogs, or whatever output he needs. Using the 9-track tape provided by the user (or purchased from MARC-0), matches are transferred to the tape. Search input may be either tape or cards. The Reference Listing shows all LC card numbers which were searched and whether there was a record printed to tape or not. Cost is 15¢ per record copied with minimum of $10.00 as of February, 1973.

Tucson Public Library is one SWLA regional library which uses this service. They are integrating the MARC-0 data with the BATAB (Baker and Taylor Automated Bookbuying System) acquisition system to coordinate their technical processes departments and speed up the processing of materials for the shelves.

The third SLICE/MARC-0 service is the "Standard S, D, I, Current Awareness" program. As of November, 1972, standard profiles in eleven subject areas are matched against the weekly MARC tape and the output can be used for current awareness, book selection and/or cataloging
information. A sample output of the Library and Information Science profile is seen below in Figure 4. Costs are determined by weekly average number of items in the bibliography and number of subscribers. For example, Library and Information Science costs $78.00 per year in 1972. This service is being discontinued at this time due to cuts in the federal funding available to the Department (Stumpff 1974, p. 1).

The Bierman and Blue article (1970a) shows the actual profile being matched for the Library and Information Science searches at that time as well as the structure to accomplish the matches. Tables are set up for LC class number and Dewey number to be matched with the key indicating which profile is being matched. Then each MARC-0 record is scanned and the appropriate fields formatted in compatible form and checked against the tables. Accession number of the record to be printed and profile number are written to an output file as a profile is matched. Records are then formatted for printing, as seen in Figure 4.

For example, the Library and Information Science search profile includes the range of numbers Z1-Z1000, 020-029, 090-099, 658.809655. Bierman would have preferred a weighted search including subject heading terminology as described below but felt that the costs involved and the
Figure 4. MARC-O S. D. I. Entry Sample
storage requirements for accessing appropriately the subject terminology did not make it feasible at that time (Bierman and Blue 1970a, p. 314). The main weakness in the search strategy is identification of a very specific subject which may be subsumed in a broader classification number. However, he concludes that the major subject areas which are reflected in the standard profiles were the most efficient way to input requests and search them, and that the results were effective in fulfilling users' needs, i.e., subscriber feedback was favorable.

The University of Arizona has subscribed to the standard profiles for the Southwest, Library and Information Science and Indians of North America. The Acquisitions staff has found them useful adjuncts to the standing order and form selection techniques predominantly used. The major question is the cost per citation received, which effectively adds $1.00 to the cost of items purchased via this selection device.

The fourth SLICE/MARC-0 service is a tailor-made profile with output similar to the third service. In this case, the user develops, with the aid of Oklahoma staff, an individual profile which may be modified several times until the weekly output is satisfactory in scope. The user indicates subject areas, Dewey Decimal classification numbers or ranges of numbers, and LC classification numbers
or ranges of numbers in defining his interest area. The
cost averages $150,000 per year but is variable depending
on the number of items in the bibliography. This presum­
ably is also discontinued, although not specifically
mentioned in the Stumpf letter.

Indiana University Student Project

William J. Studer, a doctoral candidate in the
Graduate Library School at Indiana University, conducted
research independently of the Indiana Pilot Project and
developed an SDI service for 40 faculty members in the
social sciences to alert them to monographic materials
appearing in their fields. This was considered to be of
particular interest in the nontechnical fields since book
materials are still one of the prime informational sources
in these areas as opposed to the sharp rise in interest in
the scientific fields in journal publication of research
results. As reported in the final report of the MARC Pilot
Project (Avram 1968a, pp. 179-183), 40 faculty members in
anthropology, business, economics, education, government,
history, and sociology were randomly selected from 209
interested staff. Interest profiles on each participant
were reviewed and revised; the profiles were then trans­
lated into the standardized subject headings and subject
classification terms used by LC to index the materials.
The programs were written in FORTRAN 63 for the CDC 3400/3600 at Indiana and processed the materials in three stages: extraction of new titles on the MARC master file; matching the extracted records to the profile terms and creating a tape of accession numbers and user profile number matches; and printing the appropriate MARC record(s) on three-part paper for distribution to the faculty member. One copy of the form was to be returned to Studer for evaluation purposes, one was for the faculty member to do with as he pleased, and the third was to be sent to the library if the faculty member wished to recommend it for addition to the collection. For evaluation purposes, the faculty member indicated four pieces of information: (1) whether he would recommend the title to the library or not; (2) if it matched his interest profile (yes, no, insufficient information available); (3) whether he planned to call it to the attention of another faculty member; and (4) whether the title was new to him or previously noted through some other medium.

Search strategies were weighted term-by-term (each term containing a maximum of 16 characters) with a minimum cutoff value for a match to be printed. While the dissertation project was incomplete at the time the Pilot Project final report was issued, the preliminary information suggested that the system was viable if university
administration was willing to fund SDI activities, that 75% effectiveness was reached in identifying new materials, and over half of the citations were relevant to faculty research interests. The most pleasing aspect to the faculty members involved was the convenience of making recommendations to the library for additions to the collection.

Other Projects

Brief mention should be made of three other projects: Lehigh University, Air Force Cambridge Research Laboratory, and Syracuse University School of Library Science.

Lehigh University, Bethlehem, Penna., is the only CDC 6400 user among the 54 subscribers to the MARC Distribution Service tapes. The MARC II file is used as the basis of their on-line cataloging system and serves as the core of its library automation project (Hillman 1973). The Lehigh facility stores and processes the entire MARC II data base, maintaining it as a character-oriented system for economy of storage in the process. The computer facility at Lehigh uses the ALA/USASI standard print chain, thus affording them the entire character set of the MARC II system (Davis 1973).

The Air Force Cambridge Research Laboratory uses computers to produce machine-readable files of cataloging
data and computer-produced catalog cards. They are also a subscriber to MARC II, and Markuson et al. (1972, pp. 171-175) indicate the Laboratory is currently converting programs to a cataloging module on the CDC 6600 system, in on-line mode of operation. The projected system is intended to provide book catalogs in shelflist subject, author-title, and system register number order. The database is expanded MARC, including locally produced, compatible entries.

Beginning in August, 1968, the School of Library Science at Syracuse University had the MARC Pilot tapes available for use in the Library Education Experimental Project (LEEP). The LEEP focus was twofold: "(1) the development of the laboratory for use by faculty and students, and (2) explorations of the applications in library education (Atherton and Tessier 1970, p. 25)." Five different files (including the entire MARC I data base) with supporting software were available to students in support of homework assignments and independent study units.

Sample projects are included in the 1970 article cited above. One project required the searching of a shelflist and verification that the title being classified would be in proper relationship to other items already classified in that portion of the shelflist. In addition, the subject headings which had been previously supplied in
that area were analyzed to validate consistency with the classification.

LEEP supported courses in Reference and Book Selection as well as in Technical Services. For example, the reference assignment which had as its end result the analysis and evaluation of bibliographies in various titles began with a search of the "bibliographical note" variable field (Tag 504) of data to select titles from the MARC files which had bibliographies to evaluate. This search was designed to illustrate user impact of deliberately accessing a currently unsearchable piece of data on the catalog card. The main difficulty in the assignment was the high probability that the title found on the MARC file was not available for consultation in the Syracuse Library to complete the necessary analysis (Atherton and Tessier 1970, p. 33).

The various LEEP programs provided librarians-to-be with experience manipulating machine-readable data and consequently they were given the opportunity to better evaluate when and how such innovations can be integrated effectively into the traditional library methods and resources.
CHAPTER 4

MARCA CAPABILITIES

The MARCA system has been developed at The University of Arizona to provide the students of the Graduate School of Library Science with a laboratory similar to the LEEP programs used with the MARC Pilot Project tapes at Syracuse which were described in Chapter 3. The following pages describe in some detail the programs and their limitations which combine to form the MARCA system.

The MARCA system has three programs (MARCTRS, MARCSET, and MARCIND) which are written in FORTRAN EXTENDED language with COMPASS and FORTRAN EXTENDED subroutines for use on the CDC 6400 SCOPE Operating System 3.4 at The University of Arizona. Through these three programs, the weekly MARC magnetic tapes for monographs can be successfully used to provide broad subject access to the basic catalog card information via the LC classification number. Chapter 5 will give the details on application with one particular class of graduate students.

MARCTRS

The first program in the MARCA system, MARCTRS, translates a 7-channel tape written in odd parity at 566 bpi
in the American Standard Code for Information Interchange (ASCII) character set as extended for library use to incorporate Greek characters, subscripts, superscripts, and various punctuation marks into the standard CDC character set on a character-by-character basis. The input is a stranger tape which must be EXAMINE'd past the first tape mark to reach the catalog data records. The output is a standard labelled SCOPE tape.

The LC magnetic tape has three 64-character alphabets to be translated, plus the special shift to accommodate the Greek characters and subscripts and superscripts. The standard 6-bit character set used on the tape contains the lower-case letters, numerals 0-9, standard punctuation marks, and three shift characters in positions 59, 61 and 62. Within program MARCTRS the array NEW contains 64 characters to correspond with this standard set. The following characters are lost in translation: the exclamation point, quotation marks, number sign, percent sign, and question mark. The apostrophe becomes a plus sign to enable the user to distinguish between plurals and possessives in using the output. To accommodate the special shift characters various arrows available on the CDC are used: an arrow pointing up shifts the next character to the nonstandard set I, an arrow to the right shifts the next character to nonstandard set II, and an arrow down
shifts translation of the next character to character set III (not currently used on the LC tapes).

Nonstandard character set I for the tapes consists primarily of the capital letters and various printer controls. Five punctuation marks, the field delimiter, commercial at sign (@), right and left brackets, and the reverse slash are also part of this character set. Array NEWA provides blanks for most of the carriage control characters; the shift sequence character is the only one which translates to a non-blank character: an arrow down. The field delimiter is translated to a not-equal sign, the @ symbol becomes a blank, and the reverse slash is made a regular slash.

Array NEWB contains the 64 characters in non-standard set II. This set contains various special characters such as the Icelandic thorn, the Polish 1, and Scandinavian o in both upper- and lower-case forms; and diacritical/punctuation marks such as the umlaut, miękgię znak, alif, 'ayn, and musical flat sign. Where a single character could be inserted without serious distortion of meaning, e.g., an i for the Turkish i, an e for the umlaut, a t for the Icelandic thorn, this was done. All other marks are translated to a blank.

The primary decision made in creating this program was to accept a one-to-one correspondence only in
accomplishing the translation. This minimized the storage allocation requirements of the program which is currently 75K. Each physical record read into the program has a maximum of 2048 characters. Substituting a character chain for a single character would necessitate establishment of another output array which must provide for an unknown factor above 2048 characters. Only four signs or characters not already convertible would be possible in such a system: flat, eth, plus or minus, and the th diphthong for the Icelandic thorn rather than the t currently used. The additional program coding to convert directory elements to integer form, re-figure the elements for the field involved and the succeeding elements and store the appropriate converted string would also increase the program beyond reasonable bounds to accommodate these four characters.

MARCSET

MARCSET is the second program in the series. This is the program which establishes the searchable data base of abbreviated MARC records by combining the two weekly MARC tapes and eliminating records which are corrections or deletions from earlier weekly tapes or larger than 2048 characters in the logical record. Correction and deletion records were eliminated from the translated tapes since they do not contain the complete MARC file record. The
records for items larger than 2048 characters were eliminated since the one such record on the first tape received was full of extraneous characters from earlier tape records as part of the second physical record read in. Valid and invalid characters were not separable in that record. The decision was to eliminate them from the data file but to print them out in toto to provide for analysis and consideration for later inclusion into the system. Each of the two test tapes contained one such record; a total of 4077 records are included in the sample data base for MARCSET processing.

Program MARCSET reads into a buffer the previously translated records. The appropriate Leader character is interrogated to eliminate the correction and deletion records. If the record is acceptable, the character count is converted to integer form and checked against the limit of 2048 characters. If it is a logical record larger than 2048 characters, i.e., more than one physical record for the catalog card, the total record is printed out and another record read in.

If the record is in the acceptable size range, the starting position of the variable data and the record directory elements are converted to integer form. Then several of the variable length fields are selected on the basis of tag number in the directory and processed to
eliminate the shift arrows and blanked out characters for unprintable diacritical marks. The resultant variable field contains the indicator characters, subfield delimiters and data in a variable word-length record. The final word is blank filled as necessary for completion. The record directory entry for the field is reconstituted as data tag, number of words in the field, and starting-word address within the variable fields. The resultant directory element, each portion still in integer rather than coded form, is stored in one computer word so that the right-most 15 bits contain the field tag, the next 20 bits contain the number-of-words count, and the final 25 bits store the starting-word address for the field. The chosen fields are described in full below.

The portions of the MARC record which are chosen are designed to provide the user with sufficient information on the title to correlate classification numbers in LC and Dewey classification, subject headings applied to the title and the basic bibliographic data of author, title and publisher. The internal MARC structure is also shown by including the LC card number, the fixed field data and the geographic area fields to eventually provide other searchable files.

The first variable field looked for, tag 001, contains the LC card number. This is the primary search key
for most MARC installations since it is relatively short, yet unique, identifier for the specific MARC record. This is used to print out citations for specific book titles either in standard catalog card format or for use by catalogers in setting the data into the standard card format (MARC Development Office 1972a, pp. 8-11).

The second field (tag 008) is the fixed-length data elements. From this information the user can identify the language of the text, illustration types used, intellectual level, and imprint data. There are 40 characters in this field which pack a considerable amount of data. The specific characters and their meanings are given in Table 2 (MARC Development Office 1972a, pp. 14-22). Both fields 001 and 008 are control fields, i.e., they do not contain indicator counts or subfield delimiters. As such they are a special type of variable field. All subsequent variable length fields have the introductory indicator characters, although they may not have been implemented as meaningful. All have at least one implemented subfield delimiter. The subfield delimiter on the 7-track tapes requires three characters: shift indicator to nonstandard set I, the delimiter (#), and the letter defining the specific subfield.

The third field selected, if it exists on the MARC record, is variable field 043: geographic area code. This provides one or more seven-character indicators in
<table>
<thead>
<tr>
<th>Data Element</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date entered on the MARC file</td>
<td>6</td>
</tr>
<tr>
<td>Type of imprint date</td>
<td>1</td>
</tr>
<tr>
<td>Date 1</td>
<td>4</td>
</tr>
<tr>
<td>Date 2</td>
<td>4</td>
</tr>
<tr>
<td>Country of publication</td>
<td>3</td>
</tr>
<tr>
<td>Illustration codes</td>
<td>4</td>
</tr>
<tr>
<td>Intellectual level</td>
<td>1</td>
</tr>
<tr>
<td>Form of reproduction</td>
<td>1</td>
</tr>
<tr>
<td>Microfilm</td>
<td></td>
</tr>
<tr>
<td>Microfiche</td>
<td></td>
</tr>
<tr>
<td>Microopaque</td>
<td></td>
</tr>
<tr>
<td>Large print</td>
<td></td>
</tr>
<tr>
<td>Form of contents codes</td>
<td>4</td>
</tr>
<tr>
<td>Includes bibliographies, indexes, catalogs, etc.</td>
<td></td>
</tr>
<tr>
<td>Government publication (not currently used)</td>
<td>1</td>
</tr>
<tr>
<td>Conference or meeting</td>
<td>1</td>
</tr>
<tr>
<td>Festschrift</td>
<td>1</td>
</tr>
<tr>
<td>Index indicator</td>
<td>1</td>
</tr>
<tr>
<td>Main entry in body of card</td>
<td>1</td>
</tr>
<tr>
<td>Fiction indicator</td>
<td>1</td>
</tr>
<tr>
<td>Biography indicator</td>
<td>1</td>
</tr>
<tr>
<td>Autobiography</td>
<td></td>
</tr>
<tr>
<td>Biography</td>
<td></td>
</tr>
<tr>
<td>Collective biography</td>
<td></td>
</tr>
<tr>
<td>Contains some biographical information</td>
<td></td>
</tr>
<tr>
<td>Language code</td>
<td>3</td>
</tr>
<tr>
<td>Modified record</td>
<td>1</td>
</tr>
<tr>
<td>Cataloging source</td>
<td>1</td>
</tr>
</tbody>
</table>
hierarchical order from continent to individual political entity for every geographic area connected with the book in a subject heading or when the text has a geographic orientation. The LC classification tends to disperse geographic orientations and this field may be used to serve the area specialist over the broad spectrum of subjects (MARC Development Office 1972f).

The fourth field which is selected is the LC call number: tag 050. A MARC record may have more than one 050 field but only the first is selected. The field indicators are used to show if LC has the book in its collection. The subfield delimiters are a for the class number and b for the book number. The first field will contain one or more LC classification numbers, LC call number or the word LAW for those portions of law not covered by the K schedule. Alternative LC classification numbers may be given for fiction, which LC classifies in PZ and some libraries catalog in the LC literature tables. It may also be given when LC catalogs a book in a series with the series rather than as a separate title. For example, LC may assign the call number PZ3.H318SC96, while other libraries place the same book in class PS1868. The LC catalog card and MARC records would indicate both numbers. The MARCSET translation of this field would result in: 050$aPZ3H318$S$c96$aPS1868$ on the CDC 6400 (MARC Development Office 1972a, p. 30).
The second 050 field, which MARCSET eliminates, gives a successive number applied by the cataloging library. For example, if LC had to replace a book with its microfilm edition, the first 050 field would give the original book's call number(s) and the second 050 field would give the new microfilm call number.

The fifth field, tag 082, which is transferred into the abbreviated record is the Dewey Decimal classification number. Subfield delimiter a indicates Dewey classification number. Again, alternative classification numbers are given, set off by the subfield delimiter. Slashes within the classification number indicate for the participating library which may not want the entire detail of the Dewey hierarchy where they may shorten the number. This field was selected to provide the basis for comparisons of the two major library classifications and their different viewpoints in organizing the world of knowledge (MARC Development Office 1972a, pp. 32-33).

The sixth field, tags 100-130, is the main entry field for the record. The program preserves any form of main entry: personal name, corporate name, conference, or uniform title. The second indicator always shows whether the main entry is also used as a subject heading. The first indicator varies from tag to tag in meaning. There
will be at most one 100 field per MARC record (MARC Development Office 1972a, pp. 34-39).

The seventh tag which is selected, 245, is the title statement. The first indicator in this field tells the programmer if the title is used as an added entry and the second indicator identifies up to nine characters to be ignored when using the computer to file. The subfield codes are a for the short title, b for the "rest of the title" and c for the remainder of the title page transcription: including author statement, collaborating authors, translator, etc. The entire author statement is included in the abbreviated record, although the short title would have been sufficient for bibliographic identification (MARC Development Office 1972a, pp. 42-43).

The eighth field the program isolates is tag 260: imprint field. Subfield a is the place of publication, subfield b is the publisher and subfield c is the date of publication. With tags 100-260 the basic identifying bibliographic data for the title is complete enough to use in checking the card catalog or review magazines such as Publishers' Weekly (MARC Development Office 1972a, pp. 43-44).

The final series of tags which will be included in the abbreviated record are the 600-651 entries for subject headings. This is the final information needed to complete assignments requiring comparisons of classification and
subject headings used for a segment of the LC shelflist represented by the sample tapes. The primary subfields are basic heading, geographic subdivision, period subdivision, and general subdivision (MARC Development Office 1972a, pp. 51-54). As a sample of the resultant abbreviated record, the record given in Figure 1 is shown in Figure 5.

MARCIND

The final program in the series takes the abbreviated record and sets up an index sequential file based on LC class number (subfield a in field 050) for any LC classification applied to the MARC record. Within the variable length data fields the subfield delimiters are removed prior to transferring the record to disc. For the 600 tags, the subfield delimiters are changed to 6-8 similar to the printing convention on a LC card.

Searches can be formulated by LC class letter or range of class letters, LC subclass or range of subclasses, individual LC class number or range of LC class number. Once the entire disc file has been created the search requests are read into the CPU one at a time and matched against the index. Whenever an index entry matches the request, the matching disc record is read in and printed out field by field with the appropriate introductory heading "Field (tag) is" even with the final line of data
Leader | Directory
00044NAM$a2200012$k450008AA | 0000000POA

| Control Fields
000A000/0H000C500P0]000D500PAR000E5005A9

Control Fields
| LC Card No.
000G50APC+000L5005DD000N5005H- $$$$6400844

Fixed Length Data Elements
33495888888 | 681001S1964kBBENYUBBBBBBBBBB

Variable Fields
| LC Call No.
0100ENG0BBB8BBBBBBBBBBBB | 081#aPS614#b.L75EBB

Dewey No. | Main Entry
$81#a811.5082#kKBBBF | 101#aLowenfels,Walt

| Title
er,1#d1897-1#eed,BB | 181#aPoets#bof#btoday;

| Subject Heading
[1964] #bAmerican#bpoetry#v20th#bce

Figure 5. Abbreviated MARC Record Sample

¥ = blank  Ä = record terminator
for that field. Entry words may be split from one line to another when the data field goes beyond one line or eight computer words. Once the entire record is printed, the next index entry is interrogated for a match on the search strategy. When all matches have been printed, the next search request is read and processing of it begins.

The main limitation of such a search strategy is that requested search terms must be contiguous. One cannot match S660 and S665 but not S661, S662, S663, and S664. Either two separate searches must be entered: one for S660 and one for S665; or the single search S660-S665 may be entered and the middle term matches ignored. For some subject areas this is not of interest but, as was mentioned earlier, an area specialist, e.g., Latin America, may have need for Latin American botany, zoology, geology, etc. and this could necessitate several hundred individual searches.
CHAPTER 5

MARCA IN L.S. 394

During the spring, 1974, semester, the students in Library Science 394, Advanced Cataloging, were given the first opportunity to use the MARCA data base in partial fulfillment of the course objectives to study the LC classification and automated procedures in cataloging.

Class Assignment

The students were each required to submit at least one search to be made against the MARCA data base as part of the class assignment using the LC Class H schedules. Students were to take one title which they had cataloged and classified and search that number or region of numbers which included the individual call number they had assigned and verify (a) that the book they had classified would fit into the MARCA "shelflist" without conflict, and (b) that the subject headings and subject content of the title as a whole related to other titles in that portion of the collection. This is similar to the LEEP cataloging assignment described in Chapter 3.

The portion relating to part a of the assignment required simple assurance that no other title in the
shelflist had the identical call number assigned. The second portion required the analysis of the printout which resulted from their search, particularly the subject headings and subject clues in the title, to verify that their titles fit into the appropriate portion of H selected for the search.

Thirty searches were submitted against the file. A total of 447 citations were printed for these searches. The following sections discuss the results on the basis of type of search attempted.

Search by LC Class

The broadest single request which can be submitted is for the LC class letter of one of the 22 letters used in the LC classification. The result is a listing of everything on the file within that class and its subclasses. For instance, a search on LC class M in this strategy would print out abbreviated MARC records for all titles classified in M, ML and MT, the three subclasses of class M. No one in the class asked for the LC class H search.

Search by Range of LC Classes

This type of request is a multiple request similar to that described above except that more than one complete
class is requested; for example, LC class E through LC class F. No one in the class requested this type of search strategy.

Search by LC Subclass

Four students chose the subclass search strategy. For the example discussed in LC class portion above, the equivalent of searching for class M would be three individual subclass searches: one for subclass M, one for ML and one for MT.

Two of the students requested subclass HD, with the result that they had a listing of 182 individual citations to study. Subclass HB resulted in 21 matches. Subclass PJ was also requested as a supplemental search and resulted in four matches. Since this subclass is primarily Chinese, Japanese and Arabic literatures, this response reflects the minor role of English language materials in this portion of the classification scheme. The four responses include translations of Chinese and Armenian stories and three critical works on the Arabic language and literature. A copy of this listing can be found in Appendix D.

Search by Range of LC Subclasses

Also available as a search strategy is the range of subclasses approach. For example, one might want to see only subclasses ML and MT but not subclass M to continue
the musical analogy from above. This is permissible within the MARCIND program, but no one chose that strategy for this assignment.

*Search by Individual Classification Number*

Thirteen of the students in the class requested a specific LC classification number as their search strategy. Twelve of the thirteen received the response: No matches for search. Without access to the index of call numbers available within MARCA, this would appear to be a very ineffectual search mechanism. The one individual who received a match on the strategy received one citation. This is shown in Appendix D.

*Search Range of Individual Numbers*

The sixth and final search strategy available was a range of individual classification numbers. Thirteen students also chose this strategy which was much more satisfactory as a strategy for providing citations to analyze as part of the class assignment. Only two students received the "No matches for search" message using this strategy. A majority received one or two citations (four students in each category); one received five; one received eight and one selected a range which elicited 32 responses.
Fourteen out of thirty searches which were run resulted in no citations from the MARCA file. This made it impossible for the student to complete more than half of the desired class assignment.
CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

MARCA provides the students in the University of Arizona Graduate School of Library Science with an unique opportunity to study LC cataloging and classification via the MARC tapes. Through this medium they can see not only how areas of the LC classification have changed over time but also to a limited degree how the MARC II format itself has changed over time. Working directly with printouts from MARC data base, they should have a broader understanding of how the tapes can be made to work in ways other than catalog card production.

However, the first question which is invariably asked when the programs are discussed is: Can I search by author? Following closely are requests for title, LC card number and subject heading access as means of using the MARC data base. Similarly, when Public Services staff of the University of Arizona Library were requested to identify search requirements from the MARC serials format, they indicated the following ten items in order of preference: uniform title; former titles; main entry, corporate form; main entry, meeting or conference heading; topical subject headings; language of the text; LC classification; series,
title form; index and abstracting coverage; coden; and alternative titles. Thus, further programs accessing alternative fields of MARC data should be made available. The Automation and Advanced Cataloging classes and/or students working on independent study units should be active in the determination of alternative search strategies to be made available; data fields to print for various purposes for which the materials are to be used; and formatting and translations which might utilize some of the alternatives rejected in Chapter 4 in writing the current MARCA programs. Current costs with minimum formatting and data output run 40¢ per citation, without trying to recover developmental costs. Retention of the index sequential file produced with the original MARCIND run should cut costs of searches drastically but future researchers should weigh carefully the cost/benefit ratio of making additional data fields available.

Using the same program structure as is exhibited in MARCIND, it should be possible to implement searches by LC card number (tag 001), geographic area code (tag 043), title and author-title coden as are used by OCLC, and other fields deemed necessary for instructional support. The search by variable field 043 is particularly important in the SDI provision it can make to the area specialist, e.g., Dr. Trejo in Latin American studies, which the
current LC classification searching makes difficult to handle.

Research in the area of searches by subject heading should also be done, but it involves more basic study to implement than the other four possibilities mentioned above. In particular, truncation algorithms similar to the OCLC author-title and title truncations must be developed which will uniquely identify the subject heading or headings desired yet minimize the storage space for the index being created. Some of the decisions and questions that must be resolved include the extent of searchability which we desire to make available: just the basic subject heading, or through one subdivision, or through the entire hierarchy of subdivisions, or all levels as determined by the user? Should the patron access the subdivision alone, particularly the geographic subdivisions which are possible? For example, the subject WATER-SUPPLY is used as a subdivision for specific cities and towns, e.g., TUCSON, ARIZ.--WATER-SUPPLY. How should the system provide for the user who wishes both the broad topic and the specific locality? How should this be efficiently handled for the machine? How can one minimize the false drops of related words when truncation is applied? These are just a few of the questions to be resolved prior to attempting subject heading access to the MARCA data base.
The availability of other MARC tape services also provides areas of future study concerning the adaptability of MARCA to related non-monographic files. The basic format of the various tapes (books, films, maps) is the same but the variable field information available changes considerably for each form of library material and the needs of MARC users for the various forms also vary.

The support which now exists for the full ASCII character set and the extended print train in the University Computer Center also provides an area of further research. To the author, this is a relatively minor capability; but as a demonstration for the print capabilities and possible full-card formatting, providing copy in upper and lower case and more diacritical marks would enhance the output.

The existence of the Digital Equipment Corporation (DEC) 10 computer in the University Computer Center opens up other areas of future development. One possibility is conversion of the current MARCA programs or the development of the suggested programs on the DEC 10 rather than on the CDC 6400. The on-line access to the MARCA data base would add a dimension of change to the search strategy which would give the student a chance to request further searches if, for example, the single LC classification number search he starts with returns no matches. Currently, some work is
being done by Library staff with a MARC-like serials file to investigate the possibilities of on-line serials holdings information for the University community. Similar investigations could study the feasibility of DEC-10 library science instruction.

Users of MARCA need to gain experience in the integration of the data base within classroom assignments and, particularly, the limitations which the two sample tapes forming the data base impose in defining assignments. Extension of the data base with further sample tapes or with an annual subscription to create a larger searchable file should be given consideration. Ultimately some study of the validity of this teaching tool in improvement of the students' understanding of LC classification and the MARC II format must be constructed. A controlled experiment could be constructed whereby students are divided into a group with access to the computer and MARCA programs and a control group with the more traditional methods of teaching classification and MARC format. Preferably, this would be between two library schools with a team-teaching approach so that the only variable to be tested is the use of MARCA or its derivative. The results of a standardized test would then be used to detect the presence or absence of improvement in learning by use of the computer in instruction.
APPENDIX A

MARC VARIABLE FIELD TAGS
The following list of variable field tags and their field name is derived from the 5th edition of the manual Books: A MARC Format (MARC Development Office 1972a) and its supplements 1-4 (MARC Development Office 1972b-1972e). An asterisk indicates that LC does not currently include that particular field in records distributed through the MARC Distribution Service.

### Bibliographic Control Numbers

<table>
<thead>
<tr>
<th>Tag</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>LC Card Number</td>
</tr>
<tr>
<td>011</td>
<td>Linking LC Card Number</td>
</tr>
<tr>
<td>015</td>
<td>National Bibliography Number</td>
</tr>
<tr>
<td>016</td>
<td>Linking NBN</td>
</tr>
<tr>
<td>017</td>
<td>U. S. Copyright Number</td>
</tr>
<tr>
<td>020</td>
<td>International Standard Book Number</td>
</tr>
<tr>
<td>021</td>
<td>Linking ISBN</td>
</tr>
<tr>
<td>025</td>
<td>Overseas Acquisitions Number</td>
</tr>
<tr>
<td>026</td>
<td>Linking OAN</td>
</tr>
<tr>
<td>035</td>
<td>Local System Number</td>
</tr>
<tr>
<td>036</td>
<td>Linking Local Number</td>
</tr>
<tr>
<td>040</td>
<td>Cataloging Source</td>
</tr>
<tr>
<td>041</td>
<td>Languages</td>
</tr>
<tr>
<td>042</td>
<td>Search Code</td>
</tr>
<tr>
<td>043</td>
<td>Geographic Area Code</td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

### Knowledge Numbers

<table>
<thead>
<tr>
<th>Tag</th>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>050</td>
<td>LC Call Number</td>
</tr>
<tr>
<td>051</td>
<td>Copy Statement</td>
</tr>
<tr>
<td>060</td>
<td>NLM Call Number</td>
</tr>
<tr>
<td>070</td>
<td>NAL Call Number</td>
</tr>
<tr>
<td>071</td>
<td>NAL Copy Statement</td>
</tr>
<tr>
<td>072</td>
<td>NAL Subject Category Number</td>
</tr>
<tr>
<td>080</td>
<td>UDC Number</td>
</tr>
<tr>
<td>081</td>
<td>BNB Classification Number</td>
</tr>
<tr>
<td>082</td>
<td>Dewey Decimal Classification Number</td>
</tr>
<tr>
<td>086</td>
<td>Superintendent of Documents Classification</td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>090</td>
<td>Local Call Number</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>100</td>
<td>Personal Name</td>
</tr>
<tr>
<td>110</td>
<td>Corporate Name</td>
</tr>
<tr>
<td>111</td>
<td>Conference or Meeting</td>
</tr>
<tr>
<td>130</td>
<td>Uniform Title Heading</td>
</tr>
<tr>
<td>240</td>
<td>Uniform Title</td>
</tr>
<tr>
<td>241</td>
<td>Romanized Title</td>
</tr>
<tr>
<td>242</td>
<td>Translated Title</td>
</tr>
<tr>
<td>245</td>
<td>Title</td>
</tr>
<tr>
<td>250</td>
<td>Edition Statement</td>
</tr>
<tr>
<td>260</td>
<td>Imprint</td>
</tr>
<tr>
<td>300</td>
<td>Collation</td>
</tr>
<tr>
<td>350</td>
<td>Bibliographic Price</td>
</tr>
<tr>
<td>360</td>
<td>Converted Price</td>
</tr>
<tr>
<td>400</td>
<td>Personal Name-Title (Traced Same)</td>
</tr>
<tr>
<td>410</td>
<td>Corporate Name-Title (Traced Same)</td>
</tr>
<tr>
<td>411</td>
<td>Conference-Title (Traced Same)</td>
</tr>
<tr>
<td>440</td>
<td>Title (Traced Same)</td>
</tr>
<tr>
<td>490</td>
<td>Series Untraced or Traced Differently</td>
</tr>
<tr>
<td>500</td>
<td>General Notes</td>
</tr>
<tr>
<td>501</td>
<td>&quot;Bound With&quot; Note</td>
</tr>
<tr>
<td>502</td>
<td>Dissertation Note</td>
</tr>
<tr>
<td>503</td>
<td>Bibliographic History Note</td>
</tr>
<tr>
<td>504</td>
<td>Bibliography Note</td>
</tr>
<tr>
<td>505</td>
<td>Contents Note (Formatted)</td>
</tr>
<tr>
<td>506</td>
<td>&quot;Limited Use&quot; Note</td>
</tr>
<tr>
<td>520</td>
<td>Abstract or Annotation</td>
</tr>
</tbody>
</table>
Subject Added Entries

600 Personal Name
610 Corporate Name (excluding political jurisdiction alone)
611 Conference or Meeting
630 Uniform Title Heading

LC Subject Headings

650 Topical
651 Geographic Names

Other Subject Headings

*660 NLM Subject Headings (MESH)
*670 NAL Subject Headings
*690 Local Headings - Topical
*691 Local Headings - Geographic

Other Added Entries

700 Personal Name
710 Corporate Name
711 Conference or Meeting
730 Uniform Title Heading
740 Title Traced Differently

Series Added Entries

800 Personal Name-Title
810 Corporate Name-Title
811 Conference or Meeting-Title
840 Title
APPENDIX B

MARCTRS SOURCE PROGRAM
The following pages are a source listing of the program MARCTRS. The following control card sequence is used:

```
Jobname, BN, CM60000, TP2, T600, ST8.
FTN(A, OPT=2, S=IPTEXT)
REQUEST(TAPE, HI, RO, S, VSN=xxxx)*
REWIND(TAPE)
EXAMINE(E=T)
REQUEST(TAPE0, RI, VSN=xxxx, PW=xxxx)*
REWIND(TAPE0)
LGO.
REWIND(TAPE)
RETURN(TAPE)
REWIND(TAPE0)
RETURN(TAPE0)
```

There is no printed output unless the programmer requests COPYSBF or LIST in the control card sequence. A flowchart of the program follows the program listing.

*xxxx replaces the tape number and password which will be assigned whenever a tape is placed in the UCC Tape Library.
PROGRAM MARCTRS (INPUT,OUTPUT,TAPE,TAPEO,TAPE1=INPUT,TAPE2=OUTPUT,
CTAPE3=TAPE,TAPE4=TAPEO)
C THIS PROGRAM READS IN A STRANGER TAPE FROM THE LIBRARY OF CONGRESS AND
C TRANSLATES FROM ITS MODIFIED ASCII CODE TO COC UPPER CASE CHARACTER SET.
C NO RECORD WILL BE MORE THAN 2048 CHARACTERS
C
C LNK S IS AN AREA FOR STORING TRAILING BLANKS TO FILL THE FINAL WORD OF THE
C RECORD READ IN FROM THE TAPE
C
D I M E N S I O N L N K S ( 5 )
C O M M O N / A / T A P E ( 2 1 0 ) , I N ( 2 1 0 ) , L ( 1 ) , N E W ( 6 4 ) , N E W A ( 6 4 ) , N E W B ( 6 4 ) , N F W C ( 3 )
C I A IS THE INPUT BUFFER FOR THE ORIGINAL MARC RECORD AND THE RESTORE
C RECORD FOR THE TRANSLATED RECORD.
C I N IS THE AREA FOR EACH RIGHT-JUSTIFIED CHARACTER IN THE MARC RECORD TO BE
C TRANSLATED.
C I L IS THE STORAGE AREA FOR THE LENGTH PARAMETER FOR BUFFERED RECORDS
C D A T A L N K S / 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8 , 4 0 4 0 4 0 8
C T A P E MUST BE POSITIONED AFTER THE FIRST TAPEMARK
C RECORDS ARE READ IN ONE AT A TIME FOR PROCESSING
C
C C O N T I N U E ( 3 , 1 ) ( I A ( 1 ) , I A ( 2 1 0 ) )
C I F ( U N I ' ( 3 ) ) 1 9 , 2 0 , 3 0
C C A L L R E M A R K ( 2 0 ) P A R I T Y E R R O R
C S T O P
C
C C O N T I N U E
C
C C O N T I N U E
C
C C O N T I N U E
C
C N C H A R = K / 1 2
C I F ( N C H A R . E Q . 0 ) G O T O 2 6 2
C M S K = M A S K ( 6 0 , K )
I6(1) = (I6(1), A.MSK), OR, LNSK(INCHAR)

C THIS CALLS THE SUBROUTINE TO SPLIT THE WORDS INTO THE INDIVIDUAL LETTERS

CALL UNPACK

LL=L*10

C THIS CALLS THE SUBROUTINE TO IDENTIFY WHICH CHARACTER SET IS TO BE USED

N=0

C FOR TRANSLATION.

65

C LL IS THE RANGE COUNTER LIMIT FOR CHARACTERS TO BE TRANSLATED.

DO 100 I=1,LL

C THIS LOOP WILL PICK WHICH CHARACTER SET IS BEING TRANSLATED

IF(N.EQ.0) IN(I)=40B

IF(N.EQ.1) GO TO 690

IF(N.EQ.2) GO TO 790

IF(N.EQ.3) GO TO 890

IF(N.EQ.4) GO TO 990

65 IF(IN(I).EQ.739) N = 1

IF(IN(I).EQ.75B) N = 3

75

CONTINUE

N=L

L = LL

C THIS CALLS A SUBROUTINE TO RESTORE THE CHARACTERS INTO WORDS, 10 AT A TIME.

CALL REPACK

C THIS SEGMENT TRANSFERS THE RECORD TO TAPE AND CHECKS FOR PROPER COMPLETION.

BUFFER OUT (4,1) (IA(1),IA(N))

IF (UNIT(4)) 520, 20, 30

600 N=0

65 IF(IN(I).EQ.33B) N=2

C THIS SEGMENT TRANSFORMS CHARACTERS TO SET NEWA

IN(I) = NEWA(IN(I))

GO TO 100

700 N=6

90 C THIS SEQUENCE TRANSFORMS THE STANDARD SET BUT SETS AN INDICATOR IF THE SHIFT

C SEQUENCE FOR GREEK LETTERS IS CALLED FOR.

C IF (IN(I).EQ.473) N = 4

IF(IN(I).EQ.638) N = 0

IF(IN(I).EQ.420) N = 0

95

IN(I) = NEW(IN(I))

GO TO 100

800 N = 0

C THIS SEQUENCE TRANSFORMS TO THE NON-STANDARD CHARACTER SET 2

100 IN(I) = NEW9(IN(I))

GO TO 100

C THIS TRANSFORMS THE ACTUAL GREEK LETTERS

900 IF(IN(I).EQ.738) GO TO 65

IF(IN(I).EQ.41B) IN(I) = NEMC(I)

IF(IN(I).EQ.42B) IN(I) = NEMC(I)

IF(IN(I).EQ.438) IN(I) = NEMC(I)

IF(IN(I).EQ.638) IN(I) = NEM(IN(I))

GO TO 100

520 DO 201 JA=1,210
110 C THIS CLEAR THE STORAGE AREA FOR THE NEXT RECORD TO BE READ IN.
201 IA(JA) = 0
   DO 202 JA=1,2100
202 IN(JA) = 0
   GO TO 5
   END
THIS SUBROUTINE TAKES L WORDS AND SPLITS THEM INTO L*10 CHARACTERS, RIGHT-JUSTIFIED

USE */A/  
0 322 IA BSS 210  
322 4054 IN BSSZ 2100  
4406 1 L BSS 1  
USE */  

0 000000600000000000 J006 UNPACK DATA 0  
1 5110000600 C 512000322 C SA1 IA  
2 6120000012 5130004406 C SA2 IN  
3 63330 066400 SA3 L  
4 66100 34306 SAL SB1 B0  
5 43306 066400 XA X3  
6 54620 502200001 SAM MX3 6  
7 6111000001 0512000005 SB1 B1+1  
8 6144000001 0443000000 SB2  
9 5011000001 0403000034 SB3 X3  
10 36300 STORAGE USED 31 STATEMENTS 6 SYMBOLS  
11 MODEL 73 ASSEMBLY 0.261 SECONDS 13 REFERENCES
This subroutine takes 10*L characters and puts them into L words of 10 characters each.

SETS LIMIT OF CHARACTERS TO BE HANDLED

SETS LIMIT OF CHARACTERS PER WORD

INITIALIZES LIMITS COUNTERS

CLFSRS THE STORAGE WORD

SHIFTS THE STORAGE WORD

ADD INTO THE NEXT CHARACTER

INCREMENT CHARACTER COUNTER

BRANCH IF THE FINAL CHARACTER HAS BEEN TRANSFERRED

CALL FOR NEXT CHARACTER

INCREMENT CHARACTER COUNTER

IF 10 CHARACTERS NOT TRANSFERRED CONTINUE

STORE THE WORD INTO IA

CALL FOR NEXT WORD TO BE USED FOR STORAGE

INCREMENT COUNTER FOR CHARACTERS IN WORD

WORD NOT FULL, TRANSFER CONTROL TO SHIFT IT 6

STORE FINAL WORD

SHIFT FINAL WORD 6

END

STORAGE USED 37 STATEMENTS 9 SYMBOLS

MODEL 73 ASSEMBLY 0.285 SECONDS 19 REFERENCES
THIS IS A FORTRAN EXTENDED FUNCTION TO RETURN THE UNUSED BIT COUNT.
IT IS CALLED BY A STATEMENT OF THE FORM:

\[ \text{KOUNT} = \text{KUBIT}(\text{LFN}) \]

WHERE LFN IS THE DEVICE NUMBER OR THE LEFT JUSTIFIED DISPLAY CODE NAME OF THE FILE.
THE FILE MUST BE IN S OR L FORMAT.
THE VALUE RETURNED WILL BE AN INTEGER AND ALWAYS A MULTIPLE OF 12.
IF LFN IS INVALID, A COUNT OF 60 WILL BE RETURNED FOR SCOPE 3.3.
FOR SCOPE 3.4, AN INVALID LFN CAUSES THE JOB TO BE ABORTED.

THIS VERSION OF KUBIT WORKS FOR BOTH SCOPE 3.3 AND 3.4.
MARCTRS FLOWCHART

Start

DIMENSION and initialize data array

BUFFER IN 1 PRU

Unit Check

+0

ENDFILE 4

EXIT

-1

Compute LENGTH and KUBIT

Last word complete?

NO

YES

Use UNPACK to split words into individual characters

Translate each to CDC code

Write Parity Error message

EXIT

Blank fill last word read
Use REPACK to restore record to word format

BUFFER OUT 1 PRU

UNIT CHECK

Clear storage arrays

Clear storage arrays
APPENDIX C

MARCSET SOURCE PROGRAM
The following pages contain the program listing for MARCSET. The following control card sequence is used:

```
Jobname,BN,CM60000,TP2,ST8,T600.
FTN(A,OPT=2)
REQUEST(TAPE,RO,VSN=xxxx,PW=xxxx)*
REWIND(TAPE)
REQUEST(TAPEM,RI,VSN=xxxx,PW=xxxx)*
REWIND(TAPEM)
ATTACH(KRXLIB)
LOCATE(P=CPCULX)
LDSET(LIB=KRXLIB)
LGO.
REWIND(TAPE)
RETURN(TAPE)
REWIND(TAPEM)
RETURN(TAPEM)
```

Output is an 80-column listing of those MARC records which exceed the 2048 character limit per logical record. A sample printout follows the program listing in this section. The flowchart of MARCSET follows the sample printout.

*xxxx replaces the tape number and password which will be assigned whenever a tape is placed in the UCC Tape Library.
PROGRAM MARCSETDINPUT, OUTPUT, TAPE, TAPEM, TAPE1=INPUT, TAPE2=OUTPUT,
CTAPE3=TAPE, TAPE4=TAPEM
C THIS PROGRAM TAKES DCC TRANSLATED MARC II RECORDS AND RE-FORMATS THEM
INTO A WORD ORIENTED SYSTEM RATHER THAN A CHARACTER-ORIENTED SYSTEM.
C ONLY SELECTED FIELDS OF DATA ARE REFORMATTED AND TRANSFERRED AND ONLY NEW OP
C IP INFORMATION IS TRANSFERRED. THE FINAL LIMITATION IS THAT IT MUST
C BE A MARC RECORD OF NO MORE THAN 2048 CHARACTERS TO START WITH.
COMMON/A/IA(210), IN(2100), L(1)
C IA IS THE STORAGE AREA FOR THE BUFFER IN READ. AND THE WRITE AREA FOR
C RECORDS WHICH ARE SKIPPED BY THE PROGRAM AS BEING TOO LARGE TO HANDLE.
C IN IS THE STORAGE AREA FOR THE SPLIT IA RECORDS - RIGHT-JUSTIFIED SINGLE
C CHARACTERS.
C L IS THE STORAGE LOCATION FOR THE LENGTH FUNCTION RESULT TO BE SENT TO
C REPACK AND UNPACK SUBROUTINES.
COMMON/B/MM(2), MN(300)
C MM IS THE TWO WORD ARRAY WHICH HOLDS THE CONVERTED LEADER ADDRESSES.
C MN IS THE ARRAY WHICH HOLDS THE CONVERTED RECORD DIRECTORY ENTRIES IN INTEGER
C FORM.
COMMON/C/IOUT(300), K(1), ID(300), JK(1), JL(1)
C IOUT HOLDS THE RE-CONSTRUCTED RECORD DIRECTORY.
C K HOLDS THE FIELD STARTING ADDRESSES FOR TRANSFER AMONG PROGRAMS.
C IDR HOLDS THE REFORMATTED FILE
C JK AND JL ARE COUNTERS
INTEGER REQBLK(4)
DIMENSION LNKS(1)
DATA REQBLK/4H4H4H4H/TAPE,3000000000B,6L8189A /
DATA LNKS/55555555555555555555555555/
DATA IA, IN, L/210*0, 2100*0, 1*0/
DATA MM, MN/2*0, 300*0/
DATA IOUT, K, ID, JK, JL/300*0, 1*0, 300*0, 1*1, 1*0/
C LX IS THE COUNTER FOR DETERMINING WHETHER THE SECOND TAPE HAS BEEN ADDED.
LX = 0
C THIS SEGMENT READS IN A RECORD AND CHECKS FOR COMPLETION, EOF OR PARITY ERROR.
5 BUFFER IN(3, 1) (IA(1), IA(210))
35 IF(UNIT(3)) 10, 20, 33
30 CALL REMARK(20H PARITY ERROR )
STOP
C THIS SEGMENT HANDLES THE ADDITION OF A SECOND INPUT TAPE FOR CONSOLIDATION
C OF THE SAMPLE TAPES.
20 CONTINUE
LX = LX + 1
IF(LX, 6T, 1) GO TO 24
REIND 3
CALL CPCULX(4LTAPE, 0)
45 CALL SYSX(4LREQP, 0, LOCFC(REQBLK(2)))
REIND 3
GO TO 5
C THIS SEGMENT STORES THE LENGTH OF THE RECORD READ IN
10 CONTINUE
L = LENGTH(3)
C THIS CALLS FOR THE SUBROUTINE TO SPLIT THE INPUT INTO RIGHT-JUSTIFIED,
C SINGLE-CHARACTER WORDS.
C LL STORES VALUES FROM THE LENGTH FUNCTION AND ITS MULTIPLES FOR
C SUBROUTINE TRANSFERS.
50 LL = L
CALL UNPACK
LL = L*10
C THIS SEGMENT DETERMINES FOR ELIMINATION PURPOSES ITEMS WHICH ARE CORRECTIONS
C OR DELETIONS FROM EARLIER MARC TAPES.
   IF(IN(6).EQ.038.OR.IN(6).EQ.048) GO TO 2000
C THIS SEGMENT ISOLATES THOSE PORTIONS OF THE LEADER WHICH ARE TO BE PUT INTO
C INTEGER FORMAT.
   ENCODE(10,861,MM(1)) IA(1),LNKS
501 FORMAT(45,R5)
   ENCODE(10,802,MM(2)) IN(13),IN(14),IN(15),IN(16),IN(17),LNKS
502 FORMAT(531,R5)
C THIS CONVERTS TO INTEGER FORM THE NUMBER OF CHARACTERS IN THE RECORD.
   CALL BCD9IN(MM(1),INP,1,M)
   IF(M.EQ.0) GO TO 522
70    IF(M.LE.1) GO TO 523
         WRITE(2,80A)
   GO TO 2030
523    WRITE(2,809)
   GO TO 2600
522    MM(1)=INP
C IF THE TOTAL NUMBER OF CHARACTERS IS GREATER THAN 2548, CONTROL IS TRANS-  
C FERRED TO PRINT OUT THE TRANSLATED RECORD AND THEN READ IN ANOTHER RECORD.
   IF(MM(1).GT.2048) GO TO 2201
C THIS CONVERTS TO INTEGER FORM THE STARTING ADDRESS OF THE DATA.
   CALL BCD9IN(MM(2),INP,1,M)
   IF(M.EQ.0) GO TO 622
90    IF(M.EQ.1) GO TO 573
         WRITE(2,80B)
   GO TO 2300
622    MM(2)=INP
C THIS NEXT SERIES OF COMMANDS CREATES THE MARC RECORD DIRECTORY
C THIS CONVERTS THE RECORD DIRECTORY ELEMENTS TO FORM ACCEPTED BY BCD9IN
   DO 100 J = 1,100
   IF(MN(J).EQ.0.AND.MN(J+100).EQ.0.AND.MN(J+200).EQ.0) GO TO 21
       ENCODE(10,805,MN(J)),LNKS
105   ENCODE(10,806,MN(J+100)),LNKS
106   ENCODE(10,807,MN(J+200)),LNKS
107   FORMAT(45,R7)
C THIS SEGMENT CONVERTS THE DIRECTORY ELEMENTS TO INTEGER FORM
   CALL BCD9IN(MN(J),INP,1,M)
   IF(M.EQ.0) GO TO 222
   IF(M.EQ.1) GO TO 223
92    WRITE(2,80A)
   GO TO 2000
223    WRITE(2,309)
   GO TO 2030
222 MN(J+100) = INP
   CALL 5CODIN(MN(J+200),INP,1,1)
   IF(M.EQ.0) GO TO 422
   IF(M.EQ.1) GO TO 423
   WRITE(2,804)
   GO TO 2000
   422 MN(J+200) = INP
   100 CONTINUE
   21 K= K+1
   GO TO 2001 J=1,100
   C THIS ISOLATES THE TAG 001 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
   IF(MN(J).EQ.0) GO TO 2020
   IF(MN(J).EQ.1) GO TO 66
   IF(MN(J).GT.1) GO TO 2002
   MN(K+100) = MN(K+99)
   MN(K+200) = MN(K+199)
   K = K+1
   2001 CONTINUE
   2002 KP=K
   DO 2003 J=KP+100
   C THIS ISOLATES THE TAG 008 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
   IF(MN(J).EQ.0) GO TO 2020
   IF(MN(J).EQ.8) GO TO 67
   IF(MN(J).GT.8) GO TO 2004
   MN(K+100) = MN(K+99)
   MN(K+200) = MN(K+199)
   K = K+1
   2003 CONTINUE
   2004 KP=K
   DO 2005 J=KP+100
   C THIS ISOLATES THE TAG 043 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
   IF(MN(J).EQ.0) GO TO 2020
   IF(MN(J).EQ.43) GO TO 68
   IF(MN(J).GT.43) GO TO 2006
   MN(K+100) = MN(K+99)
   MN(K+200) = MN(K+199)
   K = K+1
   2005 CONTINUE
   2006 KP = K
   DO 2007 J=KP+160
   C THIS ISOLATES THE TAG 050 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
   IF(MN(J).EQ.0) GO TO 2020
   IF(MN(J).EQ.50) GO TO 69
   IF(MN(J).GT.50) GO TO 2008
   MN(K+100) = MN(K+99)
   MN(K+200) = MN(K+199)
   K = K+1
   2007 CONTINUE
   2008 KP=K
   DO 2009 J=KP+120
   C THIS ISOLATES THE TAG 082 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
   IF(MN(J).EQ.0) GO TO 2020
   IF(MN(J).EQ.82) GO TO 70
   IF(MN(J).GT.82) GO TO 2010
   MN(K+100) = MN(K+99)
MN(K+200) = MN(K+199)
K = K+1
2009 CONTINUE
2010 KP=K
DO 2011 J=KP,100
IF(MN(J).EQ.0) GO TO 2020
C THIS ISOLATES THE TAG 100 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
IF (MN(J).GE.100.AND.MN(J).LT.200) GO TO 71
IF(MN(J).GE.200) GO TO 2012
MN(K+100)=MN(K+99)
MN(K+200) = MN(K+199)
K = K+1
2011 CONTINUE
2012 KP=K
DO 2013 J=KP,100
IF(MN(J).EQ.0) GO TO 2020
C THIS ISOLATES THE TAG 245 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
IF(MN(J).EQ.245) GO TO 72
IF(MN(J).GT.245) GO TO 2014
MN(K+100)=MN(K+99)
MN(K+200) = MN(K+199)
K = K+1
2013 CONTINUE
2014 KP=K
DO 2015 J=KP,100
IF(MN(J).EQ.0) GO TO 2020
C THIS ISOLATES THE TAG 260 AND CALLS FOR THE REFORMATTING SEGMENT OF CODE.
IF(MN(J).EQ.260) GO TO 73
IF(MN(J).GT.260) GO TO 2016
MN(K+100)=MN(K+99)
MN(K+200) = MN(K+199)
K = K+1
2015 CONTINUE
2016 KP=K
DO 2017 J=KP,100
IF(MN(J).LT.600)GO TO 74
IF(MN(J).GT.651) GO TO 2018
MN(J+100) = MN(K+99)
MN(K+200) = MN(K+199)
K = K+1
2017 CONTINUE
2018 NN=JK+3
2019 GO TO 18
2020 IF(J.EQ.1) STOP
CC THIS RECREATES THE NUMBER OF WORDS AND START ADDRESS FOR THE RESTRUCTURED
C RECORD.
2021 MM(1)=JL+JK+3
MM(2)=JK+3
C THIS TRANSFERS THE REFORMATED LEADER TO OUTPUT, IOUT(1)-IOUT(3).
2030 ENCODE(10,903,IOUT) MM(1),IA(1)
903 FORMAT(IS,R5)
ENCODE(10,904,IOUT(2)) IA(2),MM(2),IA(2)
904 FORMAT(IS,R5)
ENCODE(10,905,IOUT(3)) IA(3),LNKS
905 FORMAT(IS,R6)
N=JK+3
C THIS TRANSFERS TO IOUT THE DATA FIELDS IN IOR STARTING WITH THE START OF
C DATA FIELD ADDRESS, N.
DO 2022 J=1,NO
  IOUT(N)=IOR(I)
  N=N+1
2022 CONTINUE
N = N+JL
BUFFER OUT(4,1) (IOUT(1),IOUT(N))
IF(UNIT(4)) 2000,2200,30
C THIS CREATES A WORD-ORIENTED DATA FIELD 001 AND DIRECTORY ELEMENT NEEDED.
66 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2002
C THIS CREATES THE WORD ORIENTED DATA FIELD 008 AND DIRECTORY ELEMENT NEEDED
67 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2004
C THIS CREATES THE WORD ORIENTED DATA FIELD 043 AND DIRECTORY ELEMENT NEEDED
68 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2006
C THIS CREATES THE WORD ORIENTED DATA FIELD 050 AND DIRECTORY ELEMENT NEEDED
69 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2008
C THIS CREATES THE WORD ORIENTED DATA FIELD 082 AND DIRECTORY ELEMENT NEEDED
70 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2010
C THIS CREATES THE WORD ORIENTED DATA FIELD 100 AND DIRECTORY ELEMENT NEEDED
71 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  JK = JK+1
  K = K+1
  GO TO 2012
C THIS CREATES THE WORD ORIENTED DATA FIELD 245 AND DIRECTORY ELEMENT NEEDED
72 CALL RESOR
  MN(K*200)=MN(K*99)+MN(K*199)
  IOUT(JK*3)=MN(K).OR.SHIFT(MN(K*100),15).OR.SHIFT(MN(K*200),35)
  K = K+1
C THIS CREATES THE WORD ORIENTED DATA FIELD 260 AND DIRECTORY ELEMENT NEEDED
73 CALL RESTOR
MN(K+200)=MN(K+99)+MN(K+199)
IOUT(JK*3)=MN(K).OR.SHIFT(MN(K+100),15).OR.SHIFT(MN(K+200),35)
JK=JK+1
K=K+1
GO TO 2016
C THIS CREATES THE WORD ORIENTED DATA FIELDS 600-650 AND DIRECTORY ELEMENTS
74 CALL RESTOR
MN(K+200)=MN(K+99)+MN(K+199)
IOUT(JK*3)=MN(K).OR.SHIFT(MN(K+100),15).OR.SHIFT(MN(K+200),35)
K=K+1
JK=JK+1
GO TO 2016
C THIS CLEARS THE ARRAYS FOR NEXT READ
2000 GO 1200 I=1,210
1200 IA(I)=0
325 DO 1201 I=1,210
1201 IN(I)=0
1202 MN(I)=IOR(I)=ICUT(I)=0
K=0
L=0
JK=1
JL=0
MN(1)=MN(2)=0
GO TO 5
315 2201 N=L
C THIS SEGMENT WRITES THE RECORDS NOT USED SINCE THEY HAD MORE THAN 2048 CHARACTERS.
WRITE(2,1802) (IA(I),I=1,N)
1802 FORMAT(20X,5R10)
320 2202 GO 300 I=1,210
300 IA(I)=0
BUFFER IN (3,1) (IA(I),IA(210))
IF (UNIT(3)) 90,20.30
90 CONTINUE
325 L=LENGTH(3)
N=L
WRITE(2,1802) (IA(I),I=1,N)
2089 MN(1)=MN(1)-2048
IF (MN(1).GT.2048) GO TO 2202
330 GO TO 2000
C IF THE OUTPUT FILE IS FULL, THIS SEGMENTS TERMINATES PROGRAM
2200 ENDFILE 4
CALL EXIT
C IF THE INPUT FILE IS ENDED, THIS TERMINATES THE PROGRAM
335 24 ENDFILE 4
CALL EXIT
END
IDENT UNPACK
ENTRY UNPACK

* THIS SUBROUTINE TAKES 5 WORDS OF 10 CHARACTERS AND
* CONVERTS THEM TO L*10 WORDS OF ONE RIGHT-JUSTIFIED
* CHARACTER EACH

USE /A/

0 322 IA BSS 210
322 4064 IN 9SS 2100
4466 1 L BSS 1

USE *

0 0000000000000000 UNPACK DATA 0
1 5110000000 C SA1 IA
8120000000 C SA2 IN
2 6120000012 C SA3 L
5130004406 C SB2 10
3 6333C SB3 X3
66400 SB4 B0
4 66100 SAL SB1 B0
5 43306 SAM MX3 6
1131 SB2 A2+1 CALLS NEXT IN LOCATION
20306 LX3 6 SHIFTS ORIGINAL WORD
10633 BX3 X3*X1 SELECTS LEFT-MOST CHARACTER OF REGISTER
8X3 X3 SETS A MASK OF 6 BITS
21 BX6 X3 STORES IN APPROPRIATE LOCATION
12 SA6 A2 CALLS NEXT IN LOCATION
6 54620 5022000001 SA2 A2+1 CALLS NEXT IN LOCATION
20106 LX1 6 SHIFTS ORIGINAL WORD
7 6111000001 6512000005 NE B1,*SA1 MOVEMENTS THE COUNTER FOR CHARACTERS PER WORD
10 6144000001 6443000003 SB1 B1+1 INGREGATES THE COUNTER FOR CHARACTERS PER WORD
11 5011000001 6480000004 SB2 B1+1 INGREGATES THE COUNTER FOR CHARACTERS PER WORD
12 0400000004 SB4 B4+1 CHECKS FOR FINAL WORD SPLIT

5011000001 EQ 04,A3,UNPACK
SA1 A1+1 STARTS SPLITTING NEW WORD
46200 END SAL STARTS SPLITTING NEW WORD

46200 STORAGE USED 32 STATEMENTS 6 SYMBOLS
MODEL 73 ASSEMBLY 0.336 SECONDS 13 REFERENCES
IDENT NUMB
LIST M
ENTRY NUMB

* THIS SUBROUTINE CREATES THE DIRECTORY ELEMENTS
STARTING WITH POSITION 25 IN IN AND TERMINATES WITH
RECORD TERMINATOR OF 703

USE /A/

0 322 IA
322 4264 IN
4405 1 L

USE */

0 2 MN
2 454 MN

NUMB DATA C
1 BLD MACRO X,Y,Z
SB1 X
SB2 B2+1
EQ B2+B1,Y
LDX 6
EQ Z

1 51100000352 C
5120000002 C
2 43500
3 63410 6130000070

* THIS SEGMENT CHECKS FOR THE FIELD TERMINATOR
* THIS SEGMENT STORES THE TAG IN MNI+1 AND LEFT JUSTIFIES IT

EQ B4,B3,NUMB

66200
6 12616 AZ
7 6122000001 +
10 20666 0+421000011 +
11 20652 ABE
12 611000003

* THIS SEGMENT STORES THE LENGTH OF FIELD IN MNI+100
AND LEFT JUSTIFIES IT

13 12816 4Y
14 6122000001
46200 STORAGE USED
MODEL 73 ASSEMBLY     82 STATEMENTS      13 SYMBOLS
             0.653 SECONDS     24 REFERENCES

* THIS SEGMENT STORES THE STARTING ADDRESS OF DATA IN 18
* THE POSITION MN(I+200) AND LEFT JUSTIFIES IT. 19
THIS SUBROUTINE STORES THE DATA FIELDS CHARACTER BY CHARACTER ELIMINATING ARROWS AND SUBSTITUTING WORD DIRECTORY DATA IN PLACE OF THE ORIGINAL CHARACTER COUNTS.

NEXT LOCATION IN IOR

LIMIT OF INTRODUCTORY CHARACTERS
LIMIT OF CHARACTERS PER WORD
CTR FOR CHARACTERS PER WORD
STORES COMPLETED WORD AND CALLS FOR NEXT WORD

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
22 6111000031 C SA6 IOR+91
  76600 S81 B1+1
  66600 SX6 B0
  6350 S36 B0
23 0400000013 + S6 E0 BILL IF ONE OF THE ARROWS IS FOUND, CHECK THE NEXT WORD
24 5054000091 + SOL S45 A4+1 FOR A BLANK
  6170003055 S97 S58
25 6140000072 S84 798 IF NOT A BLANK, CHECK FOR 648
  63350 S93 X5
26 0476000040 + E0 B3,97,SAMI IF NEITHER ADD CHARACTERS TO WORD AND CONTINUE
27 0473000016 + S97 648
  12656 B0,93,AX
28 0474000001 + E0 X5+X6
29 0400000017 + S45 A4+1
30 12656 SAM 9X6 X5+X6
31 6166000001 SX6 B6+1
32 0465000034 + E0 B6,85,ROXI
  20606 LX6 6
33 0400000031 + E0 SAM
34 6122000001 ROXI S82 B2+1 IF THE END OF DATA IS REACHED BE SURE WORD IS
35 5061000455 C SA6 IOR+B1 LEFT-JUSTIFIED AND THEN STORE
5161000455 C 5161000455 C SA6 IOR+B1
36 5062000144 S81 B1+1
37 76620 SX6 B2
38 6160001132 C SX6 B1
39 5400000000 + S6 A2+100
40 0400000000 + E0 RESTOR WHEN ARROWS ARE FOLLOWED BY A BLANK, THIS CALLS THE
41 63340 5400000000 + NEXT CHARACTER IN TO CHECK
42 0673000016 + E0 B3,97,SAM
43 0400000000 + LE B3,97,AX
44 0400000000 + E0 SAM
45 46200 STORAGE USED 95 STATEMENTS 20 SYMBOLS
46 0465000000 + MODEL 73 ASSEMBLY 6.809 SECONDS 45 REFERENCES
SUBROUTINE 3C03IN (IMP, IOUT, N, M)

***** REVISED 05-03-66 *****

***** JANUARY 17, 1967 - SCOPE 2.0 *****

INP - LOCATION OF BCD WORD(s) TO BE CONVERTED
IOUT - STORAGE FOR CONVERTED WORD
N - NUMBER OF INPUT WORDS (1 OR 2)
M - COMPLETION SENTINEL

IF M = -1 ILLEGAL BCD CONVERSION
IF M = 0 SUCCESSFUL CONVERSION
IF M = +1 INPUT EXCEEDED 11 CHARACTERS

DIMENSION INP(2), ITEMP(20), JTEMP(2), IFMT(12)

IF (N=1) 21, 27, 22
22 IF (N-2) 21, 16, 21
21 M=1
RETURN
16 JTEMP(2)=INP(2)
27 JTEMP(1)=INP(1)

NUM=0
ISIGN=0
ISAVE =0
ISGNPS=0
NN =N*10
GO TO (501,502), N

501 DECODE(10, 601, JTEMP(I)) (ITEMP(J), J=1,10)
601 FORMAT (10R1)
GO TO 700

502 DECODE (20, 602, JTEMP(I)) (ITEMP(J), J=1,20)
602 FORMAT (20R1)

700 CONTINUE

GO 9 I =1,NN

951 IF (ITEMP(I)-1R1) 951, 6, 951
952 IF (ITEMP(I)-1R1) 952, 6, 952
953 IF (ITEMP(I)-1R1) 953, 6, 953
954 IF (ITEMP(I)-1R1) 954, 6, 954
955 IF (ITEMP(I)-1R1) 955, 6, 955
956 IF (ITEMP(I)-1R1) 956, 6, 956
957 IF (ITEMP(I)-1R1) 957, 6, 957
958 IF (ITEMP(I)-1R1) 958, 6, 958
959 IF (ITEMP(I)-1R1) 959, 6, 959
40 IF (ITEMP(I)-1R1) 1, 6, 1
1 IF (ITEMP(I)-1R1) 0, 10, 10
40 IF (ITEMP(I)) 2, 10, 2
2 IF (ITEMP(I)-1R1) 3, 4, 3
3 IF (ITEMP(I)-1R1) 7, 15, 7

45 IF (NUM) 17, 13, 17
13 IF (ISIGN) 38, 8, 38
26 ISGNPS =-1
6 NUM =NUM+1
GO TO 9

9 I =1,NN

951 IF (ITEMP(I)-1R1) 951, 6, 951
952 IF (ITEMP(I)-1R1) 952, 6, 952
953 IF (ITEMP(I)-1R1) 953, 6, 953
954 IF (ITEMP(I)-1R1) 954, 6, 954
955 IF (ITEMP(I)-1R1) 955, 6, 955
956 IF (ITEMP(I)-1R1) 956, 6, 956
957 IF (ITEMP(I)-1R1) 957, 6, 957
958 IF (ITEMP(I)-1R1) 958, 6, 958
959 IF (ITEMP(I)-1R1) 959, 6, 959
40 IF (ITEMP(I)-1R1) 1, 6, 1
1 IF (ITEMP(I)-1R1) 0, 10, 10
40 IF (ITEMP(I)) 2, 10, 2
2 IF (ITEMP(I)-1R1) 3, 4, 3
3 IF (ITEMP(I)-1R1) 7, 15, 7

10 IF (NUM) 17, 13, 17
13 IF (ISIGN) 38, 8, 38
26 ISGNPS =-1
6 NUM =NUM+1
GO TO 9

8 I =1
9 CONTINUE

10 IF (NUM) 17, 17, 17
39 IF (ISIGN) 38, 11, 38
38 IOUT =0
55 M =-1
RETURN
11 IOUT =0
35 M = 0
RETURN
4 IF(ISIGN) 7, 14, 7
14 ISIGN = -1
GO TO 18
15 IF(ISIGN) 7, 12, 7
12 ISIGN = 1
65 18 IF(NUM) 23, 20, 23
23 ISGNPS = 0
17 IF(NUM = ISGNPS - 1) 19, 19, 21
19 NUM = ISAVE * NUM
IF (ISAVE) 26, 26, 26
70 26 CALL SHIFTL (INPCI), JTEMP(1), N, ISAVF)
NUM = NUM - ISAVE
28 GO TO (701, 702, 703, 704, 706, 709, 710, 711, 712), NUM
701 DECODE (1, 801, JTEMP(1)) IOUT
801 FORMAT (11)
GO TO 900
702 DECODE (2, 802, JTEMP(1)) IOUT
802 FORMAT (12)
GO TO 900
703 DECODE (3, 803, JTEMP(1)) IOUT
803 FORMAT (13)
GO TO 900
704 DECODE (4, 804, JTEMP(1)) IOUT
804 FORMAT (14)
GO TO 900
705 DECODE (5, 805, JTEMP(1)) IOUT
805 FORMAT (15)
GO TO 900
706 DECODE (6, 806, JTEMP(1)) IOUT
806 FORMAT (16)
GO TO 900
707 DECODE (7, 807, JTEMP(1)) IOUT
807 FORMAT (17)
GO TO 900
708 DECODE (8, 808, JTEMP(1)) IOUT
808 FORMAT (18)
GO TO 900
709 DECODE (9, 809, JTEMP(1)) IOUT
809 FORMAT (19)
GO TO 900
100 710 DECODE (10, 810, JTEMP(1)) IOUT
810 FORMAT (110)
GO TO 900
711 DECODE (11, 811, JTEMP(1)) IOUT
811 FORMAT (111)
GO TO 900
105 712 DECODE (12, 812, JTEMP(1)) IOUT
812 FORMAT (112)
900 CONTINUE
IF (IOUT) 35, 35, 24
110 24 IF(ISIGN) 25, 35, 35
25 IOUT = - IOUT
GO TO 35
END
SUBROUTINE SHIFTL (INP, IOUT, N, M)
*** REVISED 08-22-66 ***  = BCD ZERO FILL.
*** JANUARY 17, 1967 - SCOPE 2.0 ***
INP - LOCATION OF BCD FIELD TO BE SHIFTED
IOUT - STORAGE WORD(S) FOR SHIFTED FIELD
N - NUMBER OF COMPUTER WORDS IN SHIFT PROCESS
M - NUMBER OF CHARACTERS TO SHIFT
DIMENSION INP(N), IOUT(N), ITEMP(12), IFHT(9)

10 IBLANK = 10H
IF (N) 
50:N = -N
 IBLANK = 10H0000000000
 GO TO 1
51 IBLANK = 10H
1 IF(M) 2, 3, 4
4 IC = N*10
 IF(M-IC) 6, 7, 7
7 DO 8 I = 1, N
8 IOUT(I) = IBLANK
2 IF (IBLANK - 10H ) 53, 52, 53
53 N = -N
52 RETURN
6 IW = M/10 + 1
 IPOS = M-(IW-1)*10
 IF(IPOS) 2, 9, 13
3 IW = 1
9 DO 10 I = IW, N
10 ITEMP(I) = INP(I)
15 I = 1
 DO 10 J = IW, N
 IOUT(I) = ITEMP(J)
11 I = I+1
 DO 12 J = I, N
 IF (J-N) 12, 12, 2
12 IOUT(J) = IBLANK
 GO TO 2
13 DO 14 J = IW, N
 IF(J-N) 16, 17, 2
14 IW = J
 GO TO (501, 502, 503, 504, 505, 506, 507, 508, 509), IPOS
501 ENCODE (10, 601, ITEMP(J)) INP(J), INP(K)
601 FORMAT (R9, A1)
 GO TO 14
502 ENCODE (10, 602, ITEMP(J)) INP(J), INP(K)
602 FORMAT (R8, A2)
 GO TO 14
503 ENCODE (10, 603, ITEMP(J)) INP(J), INP(K)
603 FORMAT (R7, A3)
 GO TO 14
504 ENCODE (10, 604, ITEMP(J)) INP(J), INP(K)
604 FORMAT (R6, A4)
 GO TO 14
505 ENCODE (10, 605, ITEMP(J)) INP(J), INP(K)
605 FORMAT (R5, A5)
 GO TO 14
506 ENCODE (11, 606, ITEMP(J)) INP(J), INP(K)
FORMAT (R4, A6)
GO TO 14
ENCODE (10, 607, ITEMP(J)) INP(J), INP(K)
FORMT (R3, A7)
GO TO 14
ENCODE (10, 608, ITEMP(J)) INP(J), INP(K)
FORMAT (R2, A6)
GO TO 14
ENCODE (10, 609, ITEMP(J)) INP(J), INP(K)
FORMAT (R1, A9)
CONTINUE
GO TO (701, 702, 703, 704, 705, 706, 707, 708, 709), IPOS
ENCODE (10, 601, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 602, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 603, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 604, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 605, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 606, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 607, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 608, ITEMP(N)) INP(N), IBLANK
GO TO 15
ENCODE (10, 609, ITEMP(N)) INP(N), IBLANK
GO TO 15
END
MARCSET FLOWCHART

Start

DIMENSION and initialize data arrays

BUFFER IN 1 PRU

 UNIT CHECK

+0

Greater than 1? NO

YES

Increment tape counter

REWIND 3

Mount new tape 3

REWIND 3

+1

A

ENDFILE 4

EXIT

STOP

Write Parity Error message
A

10

Compute LENGTH

Use UNPACK to split into individual characters

IN(6) = C or D?

YES

Clear all arrays

NO

ENCODE character count and start address of data

Convert with BCDBIN the character count to integer

522

YES

>2048?

B

NO

C
B
- Print contents of IA
- Clear IA
- BUFFER IN next PRU
- Unit Check
- Compute LENGTH
- Print IA
- Compute characters beyond 2048

Still YES >2048?
No
2202
2000
Convert start address of data to integer with BCDBIN.

Call NUMB to reconstitute Record Directory.

805 Convert Record Directory elements to integers.

Isolate selected tags, use RESTOR to reformat data to word-oriented file; without arrows.

2018 Recompute number of words and start address.

Recompile record into IOUT.

BUFFER OUT → D

D

2000 -1

Unit Check +1 → 30

2200 +0

ENDFILE

EXIT
APPENDIX D

MARCIND SOURCE PROGRAM
The following pages are the source program for MARCIND. The control card sequence for this program is:

```
Jobname, BN, CM75000, TP1, ST8, T700.
FTN(A, OPT=2, PL=99999)
REQUEST(TAPEM, RO, VSN=xxxx, PW=xxxx)*
REWIND(TAPEM)
ATTACH(ISLIB, ID=KC)
LIBRARY(ISLIB)
LGO.
UNLOAD(TAPEM)
REQUEST(TAPE, RI, VSN=xxxx, PW=xxxx)*
REWIND(TAPE)
REWIND(MARC)
COPYBF(MARC, TAPE)
REWIND(TAPE)
RETURN(TAPE)
DISPOSE(OUTPUT, P2=CWW, COPIES=3)
```

This control sequence is used the first time MARCIND is used with a MARCSET tape. Three copies of the output are produced: one for the teacher, one for the individuals for whom searches were run, and one for the programmer to analyze. Subsequent searches can use the file copied onto TAPE as input for an abbreviated MARCIND program. Following the program listing for MARCIND are three samples of output: search for specific classification number, search for range of classification numbers, and search for individual subclass. A flowchart for the complete MARCIND sequence follows the samples.

For the second and subsequent searches where abbreviated MARCIND is used, the following control card sequence is used:
Following the flowchart of MARCIND is a program listing for the abbreviated MARCIND search program.

*xxxx Replaces the tape number and password which are assigned whenever a tape is placed into the UCC Tape Library.
PROGRAM MARCINO(INPUT, OUTPUT, TAPEM, TAPE1 = INPUT, TAPE2 = OUTPUT, TAPE3 = CTAPEM)
C THIS PROGRAM CREATES AN INDEXED DISC FILE OF ABBREVIATED MARCII
5 RECORDS AND THEN SEARCHES THAT INDEX BY LIBRARY OF CONGRESS CLASS LETTER,
C RANGE OF LC CLASS LETTERS, SUBCLASS, RANGE OF LC SUBCLASSES, INDIVIDUAL LC
C CALL NUMBER OR RANGE OF INDIVIDUAL CALL NUMBERS WHERE THE CLASS LETTERS
C ARE LEFT-JUSTIFIED IN THE FIRST WORD COMPARED AND THE NEXT NINE CHARACTERS ARE
C CENTERED AROUND THE FIRST DECIMAL POINT IN POSITION 6 OF THE SECOND WORD.
10 IN THE SEARCH, START THE CLASS LETTERS IN COLS. 11 OR 31 OF THE INPUT CARD
C AND FOR NUMBERS OF FOUR CHARACTERS BEFORE THE DECIMAL START IN COLS. 22 AND/
C OR 42. NAME OF INDIVIDUAL FOR WHOM THE SEARCH IS DONE IS PUNCHED INTO COLS
C 1-10.
15 COMMON /A/ IA(129), IN(128Q), L(1)
C IA IS THE BUFFER ARRAY FOR INPUT FROM TAPE OF THE ABBREVIATED MARCII RECORD.
C IN IS THE ARRAY FOR STORING THE INDIVIDUAL CHARACTERS, RIGHT-JUSTIFIED, OF T
C THE TOTAL ABBREVIATED MARCII RECORD.
C L IS THE CHARACTER COUNT OF WORDS READ IN ON THE BUFFER IN AND IS PASSED TO
C SUBROUTINES FOR LIMIT COUNTER.
20 COMMON /B/ MARCO(15)
C MARCO IS THE FET FOR THE INDEXED DISC FILE.
25 COMMON /C/ IB(2), IC(45), ID(7), IBB(2)
C IB AND IBB STORE INDEX ENTRY CALL NUMBERS.
C IC STORES THE MARC II RECORD DIRECTORY WITH IC(1)-IC(15) HOLDING THE TAGS
C IC(16)-IC(30) HOLDING THE FIELD LENGTH AND IC(31)-IC(45) HOLDING THE
C STARTING ADDRESS OF FIELD RELATIVE TO THE STARTING ADDRESS OF DATA.
C ID STORES THE INITIAL THREE WORDS OF THE RECORD, I.E. THE RECORD LEADER,
C IN 7 PIECES.
30 COMMON /D/ MARCBUF(8320)
C MARCBUF STORES THE INDEX TERMS AS THEY ARE CREATED AND IS USED AS THE
C STORAGE FOR THE FINAL INDEX IN THE SEARCH PORTION OF THE PROGRAM.
35 COMMON /E/ MARCBNO(4160)
C MARCBNO STORES THE DISC ADDRESS OF THE ITEM AND IS TIED ON A ONE-TO-ONE BASIS
C TO THE MARCBUF ELEMENTS 1-4160.
40 COMMON /F/ INAME(5), ICLASS(3), INAMEP(50)
C INAME STORES THE SEARCH STRATEGY REQUESTED. IN COL. 30, AN A INDICATES SINGLE
C CLASS LETTER SEARCH, B INDICATES SINGLE SUBCLASS SEARCH, C INDICATES A
C SEARCH OF A RANGE OF CLASSES AND D INDICATED THE SEARCH FOR A RANGE OF SUB-
C CLASSES.
45 C INAMEP STORES THE RIGHT-JUSTIFIED INDIVIDUAL CHARACTERS READ INTO INAME.
C ICLASS STORES THE SEARCH LETTER IN THE LC CLASS LETTER SEARCHES.
COMMON /G/ IOT(5), IOTT(50), LN(1)
C IOT STORES THE DATA IN MARC II FIELD 50.
C IOTT STORES THE INDIVIDUAL RIGHT-JUSTIFIED CHARACTERS READ IN IOT.
C LN STORES THE LENGTH FUNCTION OF THE FIELD 50.
DATA IA, IN, L, MARCO, IB, IC, IO, IBB/128*0, 128*0, 1*0, 15*0, 2*0, 45*0, 7*0
C 2*/
DATA MARCBUF/8320*0/
DATA INAME, ICLASS, INAMEP/5*0, 3*0, 50*0/
DATA MARCBNO/4160*0/
DATA IOT, IOTT, LN/5*0, 50*0, 1*0/
C THE NEXT STATEMENT Initializes THE FET.
C MARCO = 4LMARC
C THIS OPENS THE INDEX AND Initializes IT TO ALLOCATE DISC SPACE FOR THE
55 C INDEX.
CALL ISROPE(MARCO, MARCBUF, 2328)
IF (ISRCL(MARCO) .LT. 0) GO TO 2001
DO 12357 I=1,64
12357 MARCBUF(I)=0

CALL ISWRITE(MARCO,1)
IF(ISRCL(MARCO).LT.0) GO TO 2001
CALL ISOPE(MARCO,IA,28)
IF(ISRCL(MARCO).LT.0) GO TO 2001
DO 158 I=1,64
158 IA(I)=0

C Initializes the start of data on the disc and the initial position of the
C MARCBUF and MARCBNO arrays.
C IX IS THE STARTING DISC ADDRESS FOR THE MARC RECORD TO BE STORED.
IX = 2048
C IY IS THE POSITION COUNTER FOR ARRAYS MARCBNO AND MARCBUF.
IY = 1
C IXX IS THE LINE COUNTER FOR PAGING PURPOSES.
IXX = 1
C THE FOLLOWING PORTION READS IN A RECORD FROM THE TAPE AND CHECKS FOR EOF AND
C PARITY ERRORS.
5 BUFFER IN (3,1) (IA(I),IA(128))
10 L = LENGTH(3)
20 CALL ALPHA
C THIS SEGMENT REWRITES THE INDEX AT THE FRONT OF THE DISC FILE.
157 DO 160 I=1,64
160 MARCBNO(I) = MARCBUF(I)
85 CALL ISOPE(MARCO,MARCBUF,2028)
IF(ISRCL(MARCO).LT.0) GO TO 2001
DO 159 I=1,64
159 MARCBUF(I)=MARCBNO(I)
CALL ISWRITE(MARCO,1)
IF(ISRCL(MARCO).LT.0) GO TO 2001

C CLOSES DISC FILE TO MODIFICATION.
CALL ISCLOSE(MARCO)
IF(ISRCL(MARCO).LT.0) GO TO 2001
C BRANCHES TO THE SEARCH PORTION OF THE PROGRAM.
GO TO 2800
2001 CALL REMARK (20H ISRCL ERROR )
STOP

C CREATES ID ENTRIES FOR MARC LEADER PROVIDING ACCESS TO THE
C NUMBER OF WORDS IN THE RECORD AND THE START ADDRESS OF THE DATA FIELDS.
DECODE(10,905,IA(1)> 10(1),ID(2)
905 FORMAT (15,R5)
C IF MORE THAN 127 WORDS IS READ INTO THE BUFFER AREA, THE PROGRAM SKIPS TO
C THE NEXT READ SINCE THE STORAGE ALLOCATED ON DISC IS LIMITED TO 2 PRU'S PER
C MARC ABBREVIATED RECORD.
105 IF(ID(1)>GE.128) GO TO 34
906 FORMAT(10,907,IA(2)) ID(3),ID(4),ID(5)
907 FORMAT(10,906,IA(2)) ID(3),ID(4),ID(5)
110 N = ID(4) - 4
C CREATES MARC RECORD DIRECTORY BY SPLITTING WORDS 4 THROUGH START ADDRESS OF
C DATA INTO THREE COMPONENTS EACH.
DO 26 I=1,N
IC(I) = IA(I+3).A.77777B
IC(I+15) = IA(3+I).A.377777000009
IC(I+15) = SHIFT(IC(I+15),-15)
IC(I+30) = IA(I+3).A.MASK(25)
IC(I+30) = SHIFT(IC(I+30),-35)

26 CONTINUE
DO 33 I=1,N
C SEARCHES FOR FIELD 50 IN THE RECORD DIRECTIO
IF IC(I).EQ.50 GO TO 2000
33 CONTINUE
C IF FIELD 50 IS NOT PART OF RECORD READ IN OR THE INDEX ENTRIES HAVE BEEN
C CREATED, THIS PORTION ZEROS OUT THE ARRAYS FOR NEXT BUFFER IN READ.
DO 123 JA=1,128
123 IA(JA) = 0
DO 124 JA=1,1280
124 IN(JA) = 0
DO 125 JA = 1,45
125 IC(JA) = 0
DO 126 JA = 1,7
126 ID(JA) = 0
DO 127 JA = 1,5
127 IOT(JA) = 0
128 IOT(JA) = 0
C BRANCHES TO SORT ROUTINE WHEN THE INDEX LIMIT OF 4160 WORDS IS REACHED.
IF CIY.GT.4159 ) GO TO 20
GO TO 5
C IF FIELD 50 DIRECTIO FOUND, DATA IN FIELD IS TRANSFERRED TO IOT.
2000 LN = IC(1+15)
IM = IC(I+15)+ID(4)
LMN = LN
IF (LMN.GT.5) GO TO 34
DO 500 I=1,LMN
500 CONTINUE
C WORDS IN IOT ARE SPLIT INTO INDIVIDUAL RIGHT-JUSTIFIED CHARACTERS.
CALL UNPACK2
LN = LN*10
C ROUTINE CREATES THE INDEX ENTRIES OF LC CLASS NUMBER IN IB AND IBB.
CALL INDEXE
C STORES INDEX ENTRY(S) INTO MARCBUF
MARCBUF(IY)=IB(1)
C STORES DISC ADDRESS OF RECORD.
MARCBUF(IY)=IX
MARCBUF(IY+4160) = IB(2)
C INCREMENTS COUNTER TO LIMIT MARCBUF.
IY = IY+1
IF(IY.NE.9) GO TO 2002
2010 LL = L
CALL UNPACK
LL = L*10
LM = 0
N = ID(4)-4
DO 22 I=1,N
C SEARCHES FOR ENTRIES OF SUBJECT HEADINGS IN THE RECORD DIRECTORY - 600 TAGS.
IF(IN(I).LT.600) GO TO 22
LM = (IN(I)+30)*10 + (ID(4)*10)
GO TO 23
22 CONTINUE
23 IF(LM.EQ.0) LM=LL
K=ID(4)*10
C SEARCHES FOR ENTRIES OF SUBJECT HEADINGS IN THE RECORD DIRECTORY - 600 TAGS.
IF(IN(I).LT.6Q0) GO TO 22
LM = (IN(I)+30)*10 + (ID(4)*10)
GO TO 2 3
22 CONTINUE
23 IF(LM.EQ.0) LM=LL
K=ID(4)*10
C TRANSLATES SUBFIELD DELIMITERS TO BLANKS UNLESS PART OF THE SUBJECT FIELDS.
DO 2003 I=K,LM
IF(IN(I).EQ.70B) IN(I)=IN(I+1)=IN(I+2)=55B
GO TO 2003 CONTINUE
C TRANSLATE SUBJECT HEADING SUBFIELD DELIMITERS TO BLANK DASH BLANK SEQUENCE.
DO 2003 I=LM,LL
IN(I)=IN(I+1)=IN(I+2)=55B
GO TO 2004 CONTINUE
NM = L
L = LL
C CALL REPACK
C WRITES ABBREVIATED MARC RECORD AS MODIFIED INTO THE DISC FILE AND SETS THE
C START ADDRESS FOR THE NEXT STORAGE ON DISC.
C CALL ISWEDR(MARCO,IX,ID(1))
C READS THE INDEX TO THE DISC FILE INTO HARCBUF.
C SETS DISC FILE TO READ IN A MAXIMUM OF 2 PRU'S PER READ REQUEST.
C READS THE INDEX TO THE DISC FILE INTO MARCBUF.
C SETS DISC FILE TO READ IN A MAXIMUM OF 2 PRU'S PER READ REQUEST.
C STORES IBS INTO THE INDEX OF RECORD IF AN ALTERNATIVE LC CLASS NO. EXISTS IN
C THE RECORD.
C STORES IBS INTO THE INDEX OF RECORD IF AN ALTERNATIVE LC CLASS NO. EXISTS IN
C THE RECORD.
C STORES IBS INTO THE INDEX OF RECORD IF AN ALTERNATIVE LC CLASS NO. EXISTS IN
C THE RECORD.
C restores the lc class letters into marcbuf 1-4160 and the associated disc
C address into marcbuf.  
C restores the lc class letters into marcbuf 1-4160 and the associated disc
C address into marcbuf.  
C restores the lc class letters into marcbuf 1-4160 and the associated disc
C address into marcbuf.
C IF LINE OUTPUT COUNTER = 2 THEN BRANCHES TO WRITE THE MESSAGE OF NOT A MATCH.

C READS ANOTHER SEARCH CARD AND CHECKS FOR EOF AND PARITY ERRORS.

3566 READ(15,5001)(INAME(I),I=1,5)
5001 FORMAT (5R10)

C IF (EOF(1)) GO TO 5010
5010 STOP
C SPLIT THE INAME RECORD AND REPLACES BLANKS WITH BINARY ZEROS.
5010 CALL UNPACK3

C IF THE SEARCH IS ON TOTAL CLASS NUMBER, BRANCHES.

3566 IF(INAME(3).EQ.55B) INAME(3)=0
5003 CALL REPACK3

C DETERMINES WHICH TYPE OF SEARCH IS WANTED FOR CLASS AND SUBCLASS ROUTINES.

3566 IF(INAME(3).GE.50) GO TO 6000
3566 IF(INAME(3).EQ.2B) GO TO 6002
3566 IF(INAME(3).EQ.3B) GO TO 6003

K = 1
DO 6004 I=1,4160
4400 ICLASS(I)=0
GO TO 2999
6003 K = 2

C SEARCHES FOR RANGE OF CLASS LETTERS.

DO 4401 I=1,3
4401 ICLASS(I)=0
GO TO 2999

C SEARCHES FOR RANGE OF CLASS LETTERS.

DO 6005 I=1,4160
4400 ICLASS(I)=0
4401 ICLASS(I)=0
GO TO 2999

103
C SEARCHES FOR CLASS LETTER MATCH.
   ICLASS(1) = (MARCBUF(I) .A.MASK(6))
   ICLASS(2) = (INAME(I) .A.MASK(6))
   IF (ICLASS(1).EQ. ICLASS(2)) GO TO 4200
   IF (ICLASS(1).GT. ICLASS(2)) GO TO 4400

6007 CONTINUE
   GO TO 2999
6000 WRITE(*,25002) (INAME(I), I = 1, 5)
C WRITES HEADING FOR SEARCHES ON ONE OR MORE LC CLASS NUMBERS.
5022 FORMAT (H1,20X,R10.20X,2R10,20X,2R10,/) 
   IXX = 2
   IF (INAME(4).EQ.0) GO TO 6009
   K = 5
   GO 6008 I=1,4160
C IDENTIFIES MATCHES WITHIN A RANGE OF LC CLASS NUMBERS.
   IF (MARCBUF(I).LT.INAME(2)) GO TO 6008
   IF (I(MARCBUF(I)).GE.INAME(2).AND.
      (MARCBUF(I+4160).LE.INAME(3))) .AND.
      (MARCBUF(I).LE.INAME(4)).AND.
      (MARCBUF(I+4160).GT.INAME(5)) GO TO 4200
305 IF (MARCBUF(I).GE.INAME(4)).AND.
      (MARCBUF(I+4160).GT.INAME(5))) GO TO 2999
6008 CONTINUE
   GO TO 2999
6009 K=6
   GO 6010 I=1,4160
C IDENTIFIES MATCHES FOR ONE LC CLASS NUMBER.
   IF (MARCBUF(I).EQ.INAME(2)).AND.
      (MARCBUF(I+4160).EQ.INAME(3))) GO TO 4200
   IF (MARCBUF(I).GE.INAME(2)).AND.
      (MARCBUF(I+4160).GT.INAME(3))) GO TO 2999
6010 CONTINUE
   GO TO 2999
4200 JL=I
C READS IN THE MATCHING RECORD.
   CALL ISREAD(MARCO,MARCBN0(I))
C RECREATES LEADER AND RECORD DIRECTORY.
   IF (ISRC(MARCO).LT.0) GO TO 2001
   DECODE(10,905,IA(1)) ID(1),ID(2)
   DECODE(10,906,IA(2)) ID(3),ID(4),ID(5)
   DECODE(10,907,IA(3)) ID(6),ID(7)
   N = ID(4)-4
   DO 256 I = 1,N
      IC(I) = IA(I+3),A.777777
      IC(I+15) = IA(I+3),A.377777777777777
      IC(I+30) = SHIFT(IC(I+25),-15)
      IC(I+30) = SHIFT(IC(I+30),-35)
   330 CONTINUE
   WRITE (*,2611)
C THE FOLLOWING SEQUENCE PRINTS OUT THE APPREVIATED MARC RECORD.
2611 FORMAT (1H0,20X,*HARC ABBREVIATED RECORD*)
   IXX = IXX*2
3000 DO 29 I=1,15
   IF (IC(I).EQ.0) GO TO 301
   IT = IC(I+15)
   IS = IC(I+30)+ID(4)
   IF (IC(I).GT.8) GO TO 40
   IC = IC(I)
IR=IT-IS-1
IF(IT.GT.8) GO TO 35
IXX = IXX+1
C IYY COUNTS LINES PER DATA FIELD AND FORMATS THE OUTPUT FOR PAGING
IYY = 1
IF(IXX.GT.55) GO TO 31
27 WRITE(2,262) (IA(JM),JM=IS,IR)
262 FORMAT(20X,2R10)
WRITE(2,261) ICI)
261 FORMAT(1H*,IX,* FIELD *,13, * IS *)
29 CONTINUE
301 DO 523 JA=1,128
523 IA(JA)=0
DO 525 JA = 1,45
525 IC(JA)=0
DO 526 JA=1,7
526 IC(JA)=0
I = JL
C BRANCHES TO THE APPROPRIATE SEARCH ROUTINE FOR NEXT MATCH.
GO TO (6004,6005,6006,6007,6008,6010) K
C FOR DATA FIELDS GREATER THAN TAG 088 THE FIRST 5 CHARACTERS ARE REMOVED PRIOR
C TO PRINTING
365 40 IK = IA(IS).AND.7777777778
   IA(IS) = IK
   GO TO 41
31 WRITE(2,32)
32 FORMAT(1H1)
IYY = IYY+1
GO TO 31
35 IT = IT-3
IF (IT.GT.8) GO TO 36
IXX = IXX+IYY
IF(IXX.GT.55) GO TO 31
GO TO 27
36 IYY= IYY+1
GO TO 39
360 END
THIS SUBROUTINE SETS UP THE LC CLASS NUMBER INDEX FROM FIELD 50 OF THE MARC RECORD. RESULTS ARE A LEFT JUSTIFIED CLASS AND SUBCLASS LETTERS WORD AND THE SECOND WORD CENTERED AROUND A DECIMAL IN THE 6TH POSITION.

PROCESSING STARTS WITH 6TH CHAR. CF FIELD.

INITIALIZES STORAGE LOCATION.

LIMIT FOR SEARCH FOR LST NOT LETTER.

SETS LIMIT FOR TOTAL CHARACTERS TO PROCESS.

SETS CHARACTER COUNTER.

SETS CHARACTER LIMIT.

SETS TOTAL LIMIT OF CHARACTERS.

STARTS CHECKING CHARACTERS FOR INCLUSION.

CHECKS TO SEE IF IT IS A NUMBER NOT A LETTER.

SHIFTS STORAGE WORD.

ADDS THE CHARACTER.

CALLS NEXT CHARACTER.

INCREMENTS CHARACTER COUNTER.

INCREMENTS WORD LIMIT COUNTER.

CHECKS FOR LIMIT.

IF 3 TRANSFERRED OR A NUMBER FOUND, SHIFTS THE WORD THE APPROPRIATE NUMBER OF PLACES TO LEFT-JUSTIFY THE LETTERS.

STORES THE WORD IN IR.

STORES THE WORD IN IR.

CALLS IN IR + 1.

SETS CHARACTERS BEING LOOKED FOR.

RESETS THE WORD LIMIT.

CHECKS FOR TRAILING BLANKS.

STORES THE WORD IN IR.

CALLS IN IR + 1.

SETS CHARACTERS BEING LOOKED FOR.

RESETS THE WORD LIMIT.

CHECKS FOR TRAILING BLANKS.

CALLS IN IR + 1.

SETS CHARACTERS BEING LOOKED FOR.

RESETS THE WORD LIMIT.
<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>6110000057</td>
<td>SUZIQ</td>
<td>SB1 57B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SHIFTS STORAGE REGISTER</td>
</tr>
<tr>
<td>25</td>
<td>501100001</td>
<td></td>
<td>BX2 X2+X1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ADDS IN THE NEXT CHARACTER</td>
</tr>
<tr>
<td>26</td>
<td>0476000030 +</td>
<td></td>
<td>BA1 A1+1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CALLS IN NEXT CHARACTER</td>
</tr>
<tr>
<td>27</td>
<td>0545000021 +</td>
<td></td>
<td>SB7 B7+1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INCREMENTS AND CHECKS LIMIT COUNTERS</td>
</tr>
<tr>
<td>28</td>
<td>19622</td>
<td></td>
<td>B64 B6+SUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PREPARES FOR TRANSFERRING REGISTER CONTENTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CENTERED AROUND DECIMAL IN 6TH POSITION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TO IB1(2) OR IB8(2)1</td>
</tr>
<tr>
<td>29</td>
<td>76340</td>
<td></td>
<td>BX4 X4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>30</td>
<td>7140000001</td>
<td></td>
<td>SX3 B4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PREPARES FOR TRANSFERRING REGISTER CONTENTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CENTERED AROUND DECIMAL IN 6TH POSITION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TO IB1(2) OR IB8(2)1</td>
</tr>
<tr>
<td>31</td>
<td>37443</td>
<td></td>
<td>SX4 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>32</td>
<td>0334000033 +</td>
<td>SAU2</td>
<td>SX3 B4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PREPARES FOR TRANSFERRING REGISTER CONTENTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CENTERED AROUND DECIMAL IN 6TH POSITION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TO IB1(2) OR IB8(2)1</td>
</tr>
<tr>
<td>33</td>
<td>20606</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>34</td>
<td>0304000037 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>35</td>
<td>20206</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>36</td>
<td>0314000035 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>37</td>
<td>54620</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>38</td>
<td>0400000000 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>39</td>
<td>7130000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>40</td>
<td>37443</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>41</td>
<td>0334000033 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>43</td>
<td>0314000041 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>44</td>
<td>76100</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>45</td>
<td>0334000037 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>46</td>
<td>20206</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>47</td>
<td>6177000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>48</td>
<td>0476000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>49</td>
<td>20206</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>50</td>
<td>6177000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>51</td>
<td>047600030</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>52</td>
<td>20206</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>53</td>
<td>6177000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>54</td>
<td>047600030</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>55</td>
<td>6177000001</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>56</td>
<td>37443</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
<tr>
<td>57</td>
<td>0334000021 +</td>
<td></td>
<td>SX4 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ALIGNS THE NON-EXISTANT DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTION WHICH ACTUALLY SHIFTS REGISTER IF NOT LEFT-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JUSTIFIED.</td>
</tr>
</tbody>
</table>
SUBROUTINE ALPHA
C THIS SUBROUTINE SORTS INDEX INTO ALPHABETICAL ORDER BY LC CLASS NO.
C BY COMPARING SUCCESSIVELY THE MARCBUF ENTRIES AND ALWAYS PLACING THE SMALLEST
C ENTRY IN THE HIGHER OF TWO LOCATIONS IN THE ARRAY.
COMMON /O/ MARCBUF(8320)
COMMON/E/ MARCBNO(4160)
C INN PROVIDES TEMPORARY STORAGE FOR WORD BEING MOVED LOWER INTO THE ARRAY
C WHEN IT IS TOO BIG.
DIMENSION INN(3)
DATA INN/3*0/
DO 36 1=1,4160
   IX=1
   IF (MARCBUF(I).EQ.0) GO TO 36
   IF (MARCBUF(I).EQ.0) GO TO 50
15   IF (MARCBUF(I).GE.MARCBUF(I+IX)) GO TO 39
      IX=IX+1
      IF((I*IX).GT.4160) GO TO 36
   20   GO TO 41
39   IF (MARCBUF(I).EQ.MARCBUF(I+IX)) GO TO 42
   C THIS SEGMENT ACCOMPLISHES THE NECESSARY SHIFTS.
36   INN(1) =MARCBUF(I)
      INN(2) = MARCBUF(I+4160)
      INN(3) = MARCBNO(I)
      MARCBUF(I) = MARCBUF(I+IX)
      MARCBUF(I+4160) = MARCBUF(I+IX+4160)
      MARCBNO(I) = MARCBNO(I+IX)
      MARCBUF(I+IX) = INN(1)
      MARCBUF(I+IX+4160) = INN(2)
      MARCBNO(I+IX) = INN(3)
      DO 22 JA =1,3
      INN(JA)=0
      IX =IX+1
22   35   GO TO 41
42   IF (MARCBUF(I+4160).GT.MARCBUF(I+IX+4160)) GO TO 38
      IX = IX+1
      GO TO 41
36   CONTINUE
40   C THIS OVERLAYS MARCBNO THE DISC ADDRESS OF THE RECORD AND THE CLASS LETTERS
C INTO THE SAME WORD OF MARCBUF FOR FINAL TRANSFER TO DISC.
50   DO 51 JK=1,4160
       MARCBUF(JK)=((MARCBUF(JK).A.777777777777778) .OR. (MARCBNO(JK)
       C.A.777777777777778))
   51   CONTINUE
      RETURN
END

REGISTER ALLOCATION
2 REGISTERS ASSIGNED OVER THE LOOP BEGINNING AT LINE 42
IDENT UNPACK
ENTRY UNPACK

**
**
**

THIS SUBROUTINE TAKES L WORDS IN IA AND SPLITS THEM INTO 10 RIGHT-JUSTIFIED INDIVIDUAL CHARACTERS IN ARRAY NAMED IN

USE /A/

0 200 IA 9SS 1280
260 2400 IN BSSZ 1280
260 1 L BSS 1

USE *

0 000000000000000000000000000000000 UNPACK
1 5110000000 C
5120000200 C
2 6120000012 5130002600 C
3 63330
66400
4 66100
5 43306
11331
20306
10633
6 56620
502200001 20106
7 611100001
0512000005 *
10 6144000001
0443000000 *
11 501100001
040000004 *
12 46200

STORAGE USED 32 STATEMENTS 6 SYMBOLS
MODEL 73 ASSEMBLY 0.350 SECONDS 13 REFERENCES
IDENT REPACK
ENTRY REPACK
USE /A/

REPACK DATA 0
INITIALIZES ARRAYS

SA1 IA

SA2 IN

SA3 L
SET CHARACTER LIMIT

SB3 X3

SB2 10
SETS WORD COUNT

SB4 BO
INITIALIZES COUNTERS

USE *

USE *0 200 IA BSSZ 128
200 2433 IN BSS 1250
2600 1 L BSS 1

0 00000000000000000000
REPACK DATA 0

1 5110000000 C
512000200 C

2 5130002600 C
63330

3 6120000012 66400

4 66100 SAL SB1 10
76300

5 20306 12332
6144000001

6 0443000012 +
502200001

7 6111000001 0512000005 +

10 10633 54610
5011000001

11 0400000004 +

12 6111000001 0512000014 +

13 10633 54610
0400000000 +

14 23306 BILL2 LX3 6

15 0400000012 +

END

46200 STORAGE USED 35 STATEMENTS 9 SYMBOLS
MODEL 73 ASSEMBLY 0.346 SECONDS 19 REFERENCES
This subroutine takes 10*NCHARACTERS and puts them into LN WORDS OF 10 CHARS EACH.

USE   \$S
     BSS   5
     62    50
     67    1
     USE   *

000000000000000000000000 UNPACK2 DATA 0
  5110000000       C
  612000000012    C
  63330            66400
  4610000000012    C
  54306            11331
  54620            20306
  50220000001      20106
  611000001        2512000005 +
  61440000001      0433300000 +
  50110000001      0400000004 +

46200 STORAGE USED  31 STATEMENTS  6 SYMBOLS
MODEL 73 ASSEMBLY  0.328 SECONDS  13 REFERENCES
THIS SUBROUTINE TAKES 5 WORDS AND SPLIT THEM INTO 50 INDIVIDUAL RIGHT-JUSTIFIED CHARACTERS
IDENT REPACK3
ENTRY REPACK3
USE */F/
0  5 INAME BSS 5
5  3 ICLASS BSS 3
10 62 INAMEP BSS 50
USE *

0 0000000000000000 C REPACK3 DAIA 0
1 5110000000 C RESTORES TO INAME THE 5 WORDS FROM THE
2 61300000062 INAMEP 50 CHARACTERS IN INAMEP
3 66400 5110000000 C SA1 INAME
4 66100 6120000012 C SA2 INAMEP 50 CHARACTERS IN INAMEP
5 76300 6120000010 C SB3 50 SETS CHARACTER LIMIT
6 20306 6140000001 S 5120000010 C SB2 10 SETS LIMIT OF CHARACTERS PER WORD
7 12332 5022000001 S 5120000012 S SB4 80 SETS INITIAL VALUES IN COUNTERS
8 6110000001 + 0512000005 81 + 1 SET NEXT STORAGE WORD INTO REGISTER
9 10633 5011000001 S 54610 81 + 1 CALL FOR CHECK OF COUNTER
10 54610 0400000004 + 81 + 1 CALL FOR NEXT CHARACTER TO STORE
11 0400000004 + BILL 81 + 1 CHECK WORD LIMIT
12 6110000001 + BILL3 81 + 1,B2,BILL2 IF WORD LIMIT REACHED CONTINUE,
13 10633 54610 05120000014 + BILL2 81 + 1,B2,BILL2 IF WORD LIMIT REACHED CONTINUE,
14 20306 040000000012 + BILL2 LX3 6 SHIFT WORD TO LEFT-JUSTIFY RESULT
15 END

45200 STORAGE USED 34 STATEMENTS 9 SYMBOLS
MODEL 73 ASSEMBLY 0.250 SECONDS 18 REFERENCES
MARC ABBREVIATED RECORD

FIELD 1 IS 7162289
FIELD 8 IS 71128151971 NYU B Q3033 ENG
FIELD 50 IS HX833 .C34 1971B
FIELD 82 IS 320.5/7
FIELD 100 IS CARTER, APRIL.
FIELD 245 IS THE POLITICAL THEORY OF ANARCHISM.
FIELD 260 IS NEW YORK, HARPER ROW [1971]
FIELD 650 IS ANARCHISM AND ANARCHISTS.
FIELD 1 IS JUASZ M J
FIELD 8 IS HV 1571 HV
FIELD 50 IS MARC ABBREVIATED RECORD
FIELD 82 IS FIELD 1 IS 77323231
FIELD 8 IS FIELD 8 IS 7110851967 MAUAK 8 60000 ENG
FIELD 5 IS FIELD 5 IS HV1701 .M36
FIELD 82 IS FIELD 82 IS 362.4/1
FIELD 110 IS FIELD 110 IS MASSACHUSETTS INSTITUTE OF TECHNOLOGY. SENSORY AIDS EVALUATION AND DEVELOPMENT CENTER.
FIELD 650 IS FIELD 650 IS BLIND, APPARATUS FOR THE.
MARCIND FLOWCHART

Start

DIMENSION and initialize data arrays

Open disc file

BUFFER IN 1 record

5

Unit Check

20 Alphabetize index entries

+0

1

A

Rewrite index to disc file

30 Write Parity Error message

C

Close file

STOP
10

Compute LENGTH

Decode Leader elements

L>128?

Reformat Record Directory

Find tag 050?

Clear arrays

IY>4159?

YES

YES

NO

NO

YES

B

34
Find field length of field 050

L(050) > 5?

NO

Split field 050 into individual characters by UNPACK2

Use INDEXE to create LC class numbers for index

Use UNPACK to split entire record to characters

Remove subfield delimiters

Use REPACK to make 10-character words

Write changed record to disc

Increment disc start address counter
Restore disc address to MARCBNO

Open disc file for record read

Reopen disc file

Read index into core

Open disc file for record read

Restore disc address to MARCBNO

Clear INAME

IF IXX=2? YES

Print no results message

NO

D

D
Read a card

Last card read? YES

5010 STOP

NO

3566 Split into individual characters with UNPACK3

Change blanks into binary zeros

Use REPACK3 to restore INAME to word format

Print search request

Select search strategy

Search index entries

Match? YES

2999 NO

E
4200
Read in matching record

Reconstruct Leader and Record Directory

Print record heading

Print record fields

Clear input area

Return to appropriate search strategy
PROGRAM MARCINDI (INPUT, OUTPUT, TAPEM, TAPE1=INPUT, TAPE2=OUTPUT, TAPE3 = GTAPEM)
C THIS PROGRAM CREATES AN INDEXED DISC FILE OF ABBREVIATED MARCII
C RECORDS AND THEN SEARCHES THAT INDEX BY LIBRARY OF CONGRESS CLASS LETTER,
C RANGE OF LC CLASS LETTERS, SUBCLASS, RANGE OF LC SUBCLASSES, INDIVIDUAL LC
C CALL NUMBER OR RANGE OF INDIVIDUAL CLASS NUMBERS WHERE THE CLASS LETTERS
C ARE LEFT-JUSTIFIED IN THE FIRST WORD COMPARED AND THE NEXT NINE CHARACTERS ARE
C CENTERED AROUND THE FIRST DECIMAL POINT IN POSITION 6 OF THE SECOND WORD.
C I.E., FOR SEARCHES OF CLASSIFICATION NUMBER (NOT INCLUDING THE BOOK NUMBER
C OR 42). NAME OF INDIVIDUAL FOR WHOM THE SEARCH IS DONE IS PUNCHED INTO COLS
C 1-10.
COMMON /A/ IA(128), IN(1280), L(1)
C IA IS THE BUFFER ARRAY FOR INPUT FROM TAPE OF THE ABBREVIATED MARCII RECORD.
C IN IS THE ARRAY FOR STORING THE INDIVIDUAL CHARACTERS, RIGHT-JUSTIFIED, OF T
C THE TOTAL ABBREVIATED MARCII RECORD.
C L IS THE CHARACTER COUNT OF WORDS READ IN ON THE BUFFER IN AND IS PASSED TO
C SUBROUTINES FOR LIMIT COUNTER.
COMMON /B/ MARCO(15)
C MARCO IS THE FET FOR THE INDEXED DISC FILE.
COMMON /C/ IB(2), IC(45), ID(7), 188(2)
C IB AND IBB STORE INDEX ENTRY CALL NUMBERS.
C IC STORES THE MARC II RECORD DIRECTORY WITH IC(11)-IC(15) HOLDING THE TAGS
C IC(16)-IC(30) HOLDING THE FIELD LENGTH AND IC(31)-IC(45) HOLDING THE
C STARTING ADDRESS OF FIELD RELATIVE TO THE STARTING ADDRESS OF DATA.
C ID STORES THE INITIAL THREE WORDS OF THE RECORD, I.E. THE RECORD LEADER,
C IN 7 PIECES.
COMMON /D/ MARCBUF(8320)
C MARCBUF STORES THE INDEX TERMS AS THEY ARE CREATED AND IS USED AS THE
C STORAGE FOR THE FINAL INDEX IN THE SEARCH PORTION OF THE PROGRAM.
COMMON /E/ MARCBNO(4160)
C MARCBNO STORES THE DISC ADDRESS OF THE ITEM AND IS TIED ON A ONE-TO-ONE BASIS
C TO THE MARCBUF ELEMENTS 1-4160.
COMMON /F/ INAME(5), ICLASS(3), INAMEP(50)
C INAME STORES THE SEARCH STRATEGY REQUESTED. IN COL. 30, AN A INDICATES SINGLE
C CLASS LETTER SEARCH, B INDICATES SINGLE SUBCLASS SEARCH, C INDICATES A
C SEARCH OF A RANGE OF CLASSES AND D INDICATED THE SEARCH FOR A RANGE OF SUB-
C CLASSES.
C INAMEP STORES THE RIGHT-JUSTIFIED INDIVIDUAL CHARACTERS READ INTO INAME.
C ICLASS STORES THE SEARCH LETTER IN THE LC CLASS LETTER SEARCHES.
COMMON /G/ IOT(5), IOTT(50), LN(1)
C IOT STORES THE DATA IN MARC II FIELD 50.
C IOTT STORES THE INDIVIDUAL RIGHT-JUSTIFIED CHARACTERS IN IOT.
C LN STORES THE LENGTH FUNCTION OF THE FIELD 50.
DATA IA, IN, L, MARCO, IB, IC, ID, 139/128*0, 1280*0, 1*0, 15*0, 2*0, 45*0, 7*0
C,C,2*0/
DATA MARCBUF/8320*0/
DATA INAME, ICLASS, INAMEP/5*0, 3*0, 50*8/
DATA MARCBNO/4160*0/
DATA IOT, IOTT, LN/5*0, 5*0, 1*0/
C THE NEXT STATEMENT INITIALIZES THE FET.
MARCO = 4LMARC
C THIS OPENS THE INDEX AND INITIALIZES IT TO ALLOCATE DISC SPACE FOR THE
INDEX.
280 CALL ISOPEN(MARCO, MARCBUF, 2028)
IF (ISRCLE(MARCO, LT(0)) GO TO 2001
DO 161 I=1,64
161 MARCBUF(I)=0
C READS THE INDEX TO THE DISC FILE INTO MARCBUF.
   CALL ISREAD(MARCO,i)
   IF(ISRCL(MARCO).LT.0) GO TO 2001
C SETS DISC FILE TO READ IN A MAXIMUM OF 2 PRU'S PER READ REQUEST.
1157 CALL ISOPE(MARCO,IA,2)
   IF(ISRCL(MARCO).LT.0) GO TO 2001
   DO 162 I=1,64
   162 IA(I)=0
   DO 2880 I=1,64
   CALL ISREAOCMARCO,II
   IF(ISRCL(MARCO).LT.0) GO TO 2001
   DO 162 1=1,64
   IA(I)=0
   DO 2880 1=1,640
   IF(MARCBUF(I).EQ.0) GO TO 2999
   C RESTORES THE LC CLASS LETTERS INTO MARCBUF 1-4160 AND THE ASSOCIATED DISC
   C ADDRESS INTO MARCBNO.
   MARCBNO(I)=MARCBUF(I).A.77777777777777
   MARCBUF(I)=MARCBUF(I).A.777777000000000000
2880 CONTINUE
C CLEARS THE SEARCH INPUT RECORD AREA.
   DO 5003 I=1,5
   5000 INAME(I)=0
   C IF LINE OUTPUT COUNTER = 2 THEN BRANCHES TO WRITE THE MESSAGE OF NOT A
   C MATCH.
   IF(IXX.EQ.2) GO TO 3700
C READS ANOTHER SEARCH CARD AND CHECKS FOR EOF AND PARITY ERRORS.
3566 READ(1,5001)(INAME(I),I=1,5)
   5001 FORMAT(5R10)
   IF(EOF(1)) 5010,5020
5010 STOP
C Splits the INAME RECORD AND REPLACES BLANKS WITH BINARY ZFROS.
   DO 5003 I=1,50
   5003 IF(INAME(P(I)).EQ.55) INAMEP(I)=0
   CALL REPACK3
   C IF THE SEARCH IS ON TOTAL CLASS NUMBER, BRANCHES.
   IF(INAME(3).GE.5B) GO TO 6000
C PRINTS SEARCH HEADING
   WRITE(2,5005) INAME(I), INAME(2), INAME(4), INAME(5)
   5005 FORMAT(1H1,20X,R10,20X,R10,20X,2R10,/) 
   IXX = 2
C DETERMINES WHICH TYPE OF SEARCH IS WANTED FOR CLASS AND SUBCLASS ROUTINES.
   IF(INAME(3).EQ.1B) GO TO 6001
   IF(INAME(3).EQ.2B) GO TO 6002
   IF(INAME(3).EQ.3B) GO TO 6003
   K = 1
   DO 6004 I=1,4160
   6004 CONTINUE
   GO TO 2999
C IDENTIFIES MATCHES IN THE RANGE OF SUBCLASS TYPE OF SEARCH.
   DO 6004 I=1,640
   C CleARS STORAGE AREA FOR CLASS LETTER SEARCHES.
   4403 DO 6004 I=1,3
   4401 ICLASS(I1)=0
   4402 CONTINUE
   GO TO 2999
C SEARCHES FOR RANGE OF CLASS LETTERS.
DO 6005 I=1,4160
   ICLASS(1) = (MARCBUF(I),A,MASK(6))
   ICLASS(2) = (INAME(2),A,MASK(6))
   ICLASS(3) = (INAME(4),A,MASK(6))
   IF((ICLASS(1),GE,ICLASS(2)),A,(ICLASS(1),LE,ICLASS(3))) GO TO 4200
   IF((ICLASS(1),GT,ICLASS(3))) GO TO 4400
6005 CONTINUE
   GO TO 2999
3700 WRITE(2,3701)
   3701 FORMAT(20X,* NO MATCHES FOR SEARCH.*)
   GO TO 3566
C SEARCHES FOR SUBCLASS MATCHES.
6002 K = 3
DO 6006 I=1,4160
   IF (MARCBUF(I),EQ,INAME(2)) GO TO 4200
   IF (MARCBUF(I),GT,INAME(2)) GO TO 2999
6006 CONTINUE
   GO TO 2999
6001 K = 4
DO 6007 I=1,4160
C SEARCHES FOR CLASS LETTER MATCH.
   ICLASS(1) = (MARCBUF(I),A,MASK(6))
   ICLASS(2) = (INAME(2),A,MASK(6))
   IF((ICLASS(1),EQ,ICLASS(2))) GO TO 4200
   IF((ICLASS(1),GT,ICLASS(2))) GO TO 4400
6007 CONTINUE
   GO TO 2999
6000 WRITE(2,5002) (INAME(I),I=1,5)
C Writes heading for searches on one or more lc class numbers.
5002 FORMAT(1H1,20X,R10,20X,2R10,20X,2R10,/)"
IXX = 2
   IF(INAME(4),EQ,0) GO TO 6009
   K = 5
DO 6008 I=1,4160
C IDENTIFIES MATCHES WITHIN A RANGE OF LC CLASS NUMBERS.
   IF((MARCBUF(I),LT,INAME(2))) GO TO 6008
   IF(((MARCBUF(I),GE,INAME(2)),A,(MARCBUF(I)+4160),GE,INAME(3))) .AND.
      C((MARCBUF(I),LE,INAME(4)),A,(MARCBUF(I)+4160),LE,INAME(5))) GO TO 4200
   IF((MARCBUF(I),GE,INAME(4)),A,(MARCBUF(I)+4160),GT,INAME(5))) GO TO 4400
6008 CONTINUE
   GO TO 2999
6009 K=6
DO 6210 I=1,4160
C IDENTIFIES MATCHES FOR ONE LC CLASS NUMBER.
   IF((MARCBUF(I),EQ,INAME(2)),A,(MARCBUF(I)+4160),EQ,INAME(3))) GO TO 4200
   IF((MARCBUF(I),GE,INAME(2)),A,(MARCBUF(I)+4160),GT,INAME(3))) GO TO 4400
6210 CONTINUE
   GO TO 2999
4200 JL=I
C READS IN THE MATCHING RECORD.
   CALL ISREAD(MARCO,MARCB)]
C RECREATES LEADER AND RECORD DIRECTORY.
   IF (ISRL(MARCO),LT,6) GO TO 2001
DECODE(10,905,IA(1)) ID(1),ID(2)
DECODE(10,906,IA(2)) ID(3),ID(4),ID(5)
DECODE(10,907,IA(3)) ID(6),ID(7)
N = ID(4)-4
00 256 I = 1,N
IC(I)=IA(I+3),A.777779
IC(I+5)=IA(I+3),A.3777777000038
IC(I+15)=SHIFT(IC(I+15),-15)
IC(I+30)=IA(I+3),A.MASK(25)
IC(I+30) = SHIFT(IC(I+3),-35)
CONTINUE
WRITE (2,2611)
C THE FOLLOWING SEQUENCE PRINTS OUT THE APPRECIATED MARCH RECORD.
2611 FORMAT(1H0,2QX,*MARC ABBREVIATED RECORD*)
IXX = IXX+2
3000 DO 29 I=1,15
IF(IC(I).EQ.0) GO TO 301
IT = IC(I+15)
IS = IC(I+30) + ID(4)
IF(IC(I).GT.6I) GO TO 35
IXX = IXX+1
C IYY COUNTS LINES PER DATA FIELD AND FORMATS THE OUTPUT FOR PAGING
IYY = 1
IF(IYY.GT.55) GO TO 31
WRITE(2,262) (IA(JM),JM=IS,IR)
262 FORMAT(20X.8R10)
WRITE(2,261) IC(I)
261 FORMAT(1H+,1X,» FIELD *,13, * IS * )
CONTINUE
301 DO 523 JA=1,128
523 IA(JA)=0
DO 525 JA =1,45
525 IC(JA)= 0
DO 526 JA=1,7
526 ID(JA)= 0
I = JL
C BRANCHES TO THE APPROPRIATE SEARCH ROUTINE FOR NEXT MATCH.
GO TO (6004,6005,6006,6007,6008,6010) K
C FOR DATA FIELDS GREATER THAN TAG 008 THE FIRST 5 CHARACTERS ARE REMOVED PRIOR
C TO PRINTING
40 IK = IA(IS) .AND. 7777777778
IA(IS) = IK
GO TO 41
WRITE(2,32)
31 FORMAT (1H0,2QX,*MARC ABBREVIATED RECORD*)
32 FORMAT (1H0,2QX,*MARC ABBREVIATED RECORD*)
33 FORMAT (20X.8R10)
34 FORMAT (1H+,1X, » FIELD *,13, * IS * )
CONTINUE
35 IYY = 1
36 IF (IYY.GT.55) GO TO 31
IXX = IXX+IYY
37 EXIT
12
GO TO 27
36  IYY= IYY+1
GO TO 39
2001 CALL REMARK (2BH ISRCL ERROR
STOP
END
IDENT UNPACK3
ENTRY UNPACK3

* THIS SUBROUTINE TAKES 5 WORDS AND SPLITS THEM INTO
50 INDIVIDUAL RIGHT-JUSTIFIED CHARACTERS

USE /*

0 5 0
5 3 I
10 62 INAME
UNPACK3
DATA

0 5110000600 C
1 512000010 C
2 612000012
613000005
3 66400 SAL SAM
5 43306
11331 20306
20633
6 54620 5022000001
20106
7 611100001 051200005 +
10 614400001 044300000 +
11 501100001 040000004 +
12 END

46200 STORAGE USED 30 STATEMENTS 6 SYMBOLS
MODEL 73 ASSEMBLY C.318 SECONDS 12 REFERENCES
IDENT REPACK3
ENTRY REPACK3
USE /F/

0 0000000000000000 REPACK3 DATA 0 RESTORES TO INAME THE 5 WORDS FROM THE
1 5110000000 NAME BSS 5
2 6130000062 INAMEP BSS 50 SETS LIMIT OF CHARACTERS PER WORD
3 66400 SAL SB1 80 SETS LIMIT OF CHARACTERS PER WORD
4 66100 INAMEP BSS 50 SETS INITIAL VALUES IN COUNTERS
5 20306 SAM SB4 80 SETS INITIAL VALUES IN COUNTERS
6 12332 6X3 X3*X2 ADD A CHARACTER TO WORD
7 6140000001 SX3 X3 + A characterize TO WORD
8 0443000012 A2+1 INCREMENT AND CHECK COUNTER
9 5020000001 SB1 B1+1 CALL FOR CHECK OF LIMIT
10 6110000001 NE B1,B2,SAM CONTINUE TRANSFERRING CHARACTERS IF LIMIT OK
11 10633 6X6 X3 STORAGE OF WORD IF LIMIT REACHED
12 54610 SB1 B1+1 CHECK WORD LIMIT
13 5010000001 NE B1,B2,BILL CONTINUE, CONTINUE, REPRESENTS TO RETURN TO CALLING PROGRAM
14 0400000014 + BILL2 L X3 6 SHIFT WORD TO LEFT-JUSTIFY RESULT
15 END 0400000012 + BILL2 L X3 6 SHIFT WORD TO LEFT-JUSTIFY RESULT

45200 STORAGE USED 34 STATEMENTS 9 SYMBOLS
MODEL 73 ASSEMBLY 6.263 SECONDS 18 REFERENCES
APPENDIX E

ACRONYMS USED IN THIS THESIS
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>BATAB</td>
<td>Baker and Taylor Automated Bookbuying System</td>
</tr>
<tr>
<td>BIBNET</td>
<td>Commercial system for on-line card production</td>
</tr>
<tr>
<td>BPI</td>
<td>Bits per inch</td>
</tr>
<tr>
<td>CARDS</td>
<td>Card Automated Reproduction and Distribution System</td>
</tr>
<tr>
<td>CDC</td>
<td>Control Data Corporation</td>
</tr>
<tr>
<td>CIP</td>
<td>Cataloging In Publication</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CRT</td>
<td>Cathode ray tube</td>
</tr>
<tr>
<td>DEC</td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td>DOS</td>
<td>Disc Operating System</td>
</tr>
<tr>
<td>EXAMINE</td>
<td>Program which gives octal or alphabetic dumps of tapes which have been produced at another installation or upon another computer, commonly called Stranger Tapes</td>
</tr>
<tr>
<td>IDC</td>
<td>Information Dynamics Corporation</td>
</tr>
<tr>
<td>ISBN</td>
<td>International Standard Book Number</td>
</tr>
<tr>
<td>LC</td>
<td>Library of Congress</td>
</tr>
<tr>
<td>LEEP</td>
<td>Library Education Experimental Program</td>
</tr>
<tr>
<td>MARC</td>
<td>MACHine-Readable Cataloging</td>
</tr>
<tr>
<td>MARCA</td>
<td>System of programs used at The University of Arizona to process MARC tapes</td>
</tr>
<tr>
<td>MARCIND</td>
<td>Program which creates sequential disc file and searches the index file; part of MARCA</td>
</tr>
<tr>
<td>MARC-0</td>
<td>Project of the Oklahoma Department of Libraries for the use of MARC tapes</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MARCSET</td>
<td>Program which creates simplified MARC records in word format for the CDC 6400; part of MARCA</td>
</tr>
<tr>
<td>MARCTRS</td>
<td>Program which performs a character-by-character translation of MARC tapes from LC to the local CDC 6400 character set; part of MARCA</td>
</tr>
<tr>
<td>MUMS</td>
<td>Multiple Use MARC System</td>
</tr>
<tr>
<td>NELINET</td>
<td>New England Library Network</td>
</tr>
<tr>
<td>OCLC</td>
<td>Ohio College Library Center</td>
</tr>
<tr>
<td>OCR</td>
<td>Optical Character Recognition</td>
</tr>
<tr>
<td>RECON</td>
<td>REtrospective CONversion project of LC to convert pre-MARC catalog cards to MARC II format via format recognition programs</td>
</tr>
<tr>
<td>SCOPE</td>
<td>Operating system of the CDC 6400</td>
</tr>
<tr>
<td>SDI</td>
<td>Selective Dissemination of Information</td>
</tr>
<tr>
<td>SLICE</td>
<td>Southwestern Library Interstate Cooperative Endeavor</td>
</tr>
<tr>
<td>SWLA</td>
<td>Southwestern Library Association</td>
</tr>
<tr>
<td>UCC</td>
<td>University Computer Center</td>
</tr>
<tr>
<td>USASI</td>
<td>United States of America Standards Institute</td>
</tr>
</tbody>
</table>
SELECTED BIBLIOGRAPHY


