THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING ON THE PAGE BEING CROOKED. THIS IS THE BEST IMAGE AVAILABLE.
AN INFORMATION SYSTEM FOR EDUCATIONAL PLACEMENT

by

RICHARD ELLIS LONSINGER

B. S., Kansas State University, 1965

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Approved by:

[Signature]
Major Professor
TABLE OF CONTENTS

LIST OF FIGURES ........................................ iv

1. INTRODUCTION ........................................ 1
   EDUCATIONAL PLACEMENT DATA PROBLEM ............... 1

2. DESIGN IMPLEMENTATION ............................... 2

3. DESCRIPTION OF DATA FILES .......................... 6
   TOTALREC (Master) File ............................... 6
   INDEXREC (Social Security Number) File ............ 6
   COMPORFN (Competencies, Organizations and Functions) File 7
   LOCRGDIR (Location Preference) File ............... 8

4. FILE INITIALIZATION .................................. 9

5. PROGRAMS FOR DATA FILES ............................ 9
   CREATE Program ...................................... 9
   DIRCREA Program ................................... 10
   FILEDEVL Program .................................. 13

6. CONCLUSIONS ......................................... 15

7. BIBLIOGRAPHY ........................................ 16

8. APPENDICES
   APPENDIX A (TOTALREC Record Variable List) ........... 17
   APPENDIX B (Inactivation Input Card Format for
                FILEDEVL Program) ........................... 21
   APPENDIX C (Inactive Input Card Format for FILEDEVL
                Program) .................................. 22
   APPENDIX D (File Activation Input Card Format for
                FILEDEVL Program) .......................... 23

ii
<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Alumnum File Activation Input Card Format for FILEDEVL Program</td>
<td>25</td>
</tr>
<tr>
<td>F</td>
<td>FILEDEVL Program Listing</td>
<td>27</td>
</tr>
<tr>
<td>G</td>
<td>CREATE Program Listing</td>
<td>41</td>
</tr>
<tr>
<td>H</td>
<td>DIRCREA Program Listing</td>
<td>47</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

FIGURE .................................................................................................................. PAGE
1. Chaining of Data Records for a Coded Directory Entry ........................................... 5
2. Input Card Format for COMPORFN File Initialization as used by the DIRCREA Program ................................................................. 11
3. Input Card Format for LOCRGDIR File Initialization as used by the DIRCREA Program ................................................................. 12
AN INFORMATION SYSTEM FOR EDUCATIONAL PLACEMENT

INTRODUCTION

The Kansas State University Career Planning and Placement Center provides service to students and alumni through its two major sections. One section serves those individuals seeking employment in educational institutions; the other serves those seeking employment in business, industry, and government. This report describes the design of an information storage and retrieval system to be used by the Educational Placement Section in providing service to those persons seeking employment in the broad field of education. With minor modifications the information system could provide assistance to all individuals using the services of the Career Planning and Placement Center.

EDUCATIONAL PLACEMENT DATA PROBLEM

The Educational Placement Section serves as a focal point for persons seeking employment in education and for educational institutions and businesses seeking employees. In performing its function a variety of detailed information must be maintained on job candidates and employers. It was determined that the most important function, and that function most difficult to continue using conventional data handling methods, was maintaining information about the individuals seeking employment.

The Education candidate registers with the Educational Placement Section usually during the fall semester. He provides a variety of information on several different forms. Thus, when Placement Center personnel desire information about job candidates, searches through the regis-
tration forms were necessary. The basic problem associated with Educational Placement manual data handling operations was that they were inefficient and inflexible. A more flexible approach to the problem was found in using information storage and retrieval techniques for handling required data. The required information on job candidates would be readily available for rapid retrieval and processing by automating portions of the information handling problem. Summary information for yearly activity reports could also be rapidly processed.

The approach taken to solve the problem uses a copy of the Admissions and Records master enrollment tape which contains most of the basic information necessary for Educational Placement registration. Use of this master tape file reduces the registration procedure considerably. To register, the registrant completes a single form which provides the Educational Placement Section with the information concerning the level he wants to teach, his areas of competence, and his location preference.

DESIGN IMPLEMENTATION

The information from the registration form is used with the information taken from the Admissions and Records master tape to construct a master Educational Placement record on a direct access storage file. If the registrant is not presently enrolled at the University, he provides the necessary master record information on the registration form. This information is punched onto cards and entered into the system to complete the construction of his master record.

Master Educational Placement file records are referenced indirectly through the use of an associated file. Each entry in the associated file consists of a social security number and the record number of the registrant's
master record within the master data file. The associated file is arranged in ascending sequence by social security numbers. Retrieval of master record information for a registrant is rapidly accomplished by using his social security number as the search argument in a binary search of the associated file. When the desired social security number matches a social security number in the file, the record number associated with the social security number is used to enter the master file at the exact location of the desired registrant information. Use of this retrieval technique does not require the master data file to be in any particular arrangement.

Additional flexibility was achieved in the information system by using two directory files. One directory contains coded information for teaching competencies and types of organizations and functions. The other directory contains coded information for job location preferences. The registrant provides the appropriate code numbers from the established Educational Placement codes when he completes his registration form.

Each directory entry consists of an abbreviation of the particular entry name and a record number. The record number is a pointer to the last master data file record entered in the system that used the entry.

In the teaching competencies, organizations and functions file, each entry is referenced by a displacement in the file equal to its code number. In the location preferences file, each entry is referenced by a displacement computed from its code number.

The code numbers specified by the registrant are stored at specific data areas in his master data record. Associated with each of his data areas are two additional data areas which are used for backward and forward chaining pointers. When a code is used in constructing a new master record, the backward chain pointer in the record being constructed is set to point
to the previous record that used the code number. The forward chain pointer in the record being created is set to zero. Since the directory for the code number points to the previous record to use the code, that record is retrieved and its forward chain pointer is set to point to the new record. The directory entry for the code number is updated to point to the new record just constructed. Thus, each record that uses a particular code number will point to the previous record and the subsequent record that used the code number.

Retrieval of registrant information by code number can be accomplished easily by entering the directory at a displacement determined by the code number and getting the record number of the last record to use the code. The retrieval is completed by traversing the backward chain pointers of all the records using the code until the null pointer in the first record is detected.

An example of the chaining technique is given in Fig. 1. In the example three records used code number 100. Master data record number 593 was the last record created. Before it was created, the directory record for code 100 contained number 427 since that was the last record to use the code (point A). The backward chain pointer in record 593 was set equal to 427 and the forward chain pointer was set equal to zero. Record 427 was retrieved and its forward chain pointer was set to equal 593. The directory record was updated to indicate that record number 593 was the last record to use code number 100 (point B).

It is seen from the example that by entering the directory as shown at point B, the retrieval of all the records that used the code number 100 can be easily accomplished.
POINT A - DIRECTORY RECORDS

```
.......
   code last record to use this code
   100   427
   .......
```

MASTER DATA FILE

```
.......
   code backward forward
       pointer   pointer

390 ....  100   0   427
    .......

427 ....  100   390  593
    .......

593 ....  100   527   0
    .......
```

POINT B - DIRECTORY RECORDS

```
.......
   code last record to use this code
   100   593
   .......
```

Figure 1. Chaining of data records for a coded directory entry.
DESCRIPTION OF DATA FILES

The information system was designed to use a total of four direct access data files. These files are the master file (system name - TOTALREC file), the associated file (system name - INDEXREC file), and the two directory files (system names - COMPORFN file and LOCRGDIR file). The system also uses the Admissions and Records master tape file for initialization of the TOTALREC and INDEXREC files.

The TOTALREC file format is shown in App. A. Each record in the file is 320 bytes long. The first 164 bytes of each record contains the information taken from the Admissions and Records master tape file or from punched cards for an alumnus. The remainder of the file contains update information taken from the Educational Placement registration form or optionally, from the Kansas State University diploma receipt. This file contains information for all students who have specified that they anticipate graduating on or before the first of September of the next academic year. The file has a capacity of 3400 records.

The INDEXREC file is created simultaneously with the TOTALREC file from the Admissions and Records master tape file. This file contains an encoded array of social security numbers and their corresponding record numbers in the TOTALREC file. Each physical record in this file is 3520 bytes long, and contains 440 encoded logical records. Each 8 bytes logical record corresponds to an element of an incore array which must be dimensioned (3400, 2) in the FORTRAN programs using this file. The array was blocked for the direct access file to conserve file space.

The decoding procedure in a processing program begins by reading a block of data from the file into an 880 byte working array. The odd num-
bered elements of the working array contain the social security numbers while the even numbered elements contain the master file record numbers. The social security numbers are put into the first dimension of each element of a two-dimensional array (3400, 2) while the master file record numbers are put into the second dimension of each element. Additional blocks are read and decoding continues until a minus one (-1) is encountered in an input block. This marks the logical end of the data and terminates the decoding procedure. The social security numbers are maintained in the index array in collating sequence to allow retrieval of their corresponding record numbers using a binary search technique. When the program finishes processing data the index array is encoded into 3520 byte blocks which are filed to the INDEXREC file.

The two directory files, COMPORFN and LOCGRGDIR, also contain encoded array information. The COMPORFN file contains information for teaching competencies, types of organizations, and functions. It was created by a special program using the code numbers provided by the Educational Placement Section. Each physical record of the file is 800 bytes long, and contains 25 encoded logical records. Each logical record contains 16 characters of descriptive information and space for four (4) link variables. Each 32 byte logical record corresponds to an element of an in-core array which must be dimensioned (1000, 8) in the FORTRAN programs using this file. The array was blocked for the direct access file to conserve file space.

The decoding procedure is identical in function to that of the INDEXREC file. Blocks of data from the file are placed into a 200 element working array and the 32 byte logical records are moved into a successive array elements. Processing continues until exactly 40 blocks of data have been processed from the file.
In this array the first dimension corresponds to the code number for
the variables. The eight (8) positions in the second dimension provide for
descriptive information about each variables and link variable information.
The link variable is used to store the record number of the last record in
the master file (TOTALREC file) that used the directory entry. Although
four (4) link levels are provided in the design, only level one was used in
the initial programs. When the program finishes processing data the updated
array is blocked into 800 byte blocks which are written to the COMPORFN
file.

The LOCRGDIR file contains the information for location preferences.
It was created by the same program that created the COMPORFN file. Each
physical record of the file is 800 bytes long, and contains 50 encoded
logical records. Each logical record contains a four (4) character state
or regional abbreviation and space for three (3) link variables (only the
first was used in the initial programs). Each 16 byte logical record is
decoded into an in-core array which must be dimensioned (550, 4) in the
FORTRAN program using the file. Decoding of this array is similar to that
of the COMPORFN file. For this file only eleven (11) blocks of data are
processed from the file.

In this array the first dimension corresponds to the code numbers for
state and regional location preferences. The four (4) positions in the
second dimension provide space for descriptive information and link varia-
ble information. The link variable is used to store the record number of
the last record in the master (TOTALREC) file that used this location pre-
ference. The array is reblocked and written to the LOCRGDIR file when pro-
cessing is completed.
FILE INITIALIZATION

The direct access file space used by the data sets should be allocated on the computing system as a separate job step. The separate allocation avoids the necessity of changing job control cards on any subsequent programs.

All System/360 direct access data sets must be initialized using the IBM supplied Basic Sequential Access Method (BSAM) before these data sets can be processed in the direct access mode. The BSAM technique accesses each record in the file sequentially starting with record number one (1). It marks the logical end of the data file with a unique record. Subsequent accesses of the data on a properly initialized file can be made of any record directly without regard to its physical sequence within the data file. Any attempt to read or write directly to the file at a point beyond the end of the logical data file will cause the program to terminate.

PROGRAMS FOR DATA FILES

CREATE Program

The master record file (TOTALREC) and the associated record file (INDEXREC) were created simultaneously by the program named CREATE (Appendix G). This program, written in the System/360 Assembler language, uses the Admissions and Records master tape file and one card containing the maximum graduation date of interest for inputs. The input card format has a two digit numeric month code in columns 1-2, and the last two digits of the year in columns 3-4.

The primary function of the CREATE program is to select parts of the Admissions and Records data and convert the select parts from one byte
character data to two (2) or four (4) byte integer data (e.g., the social security number is converted from a nine (9) byte character field to a four (4) byte binary field. The data conversion process creates a data file which can be processed using unformatted FORTRAN input/output commands.

**DIRCREA Program**

The competencies, organization, and functions file (COMPORFN) and the location and regions file (LOCRGDIR) are initialized by the DIRCREA program (Appendix H). This program, written in FORTRAN IV, uses the Educational Placement codes for input. The actual code numbers are not stored in the files, but are used to determine the relative position of the particular data item within the files.

The input data for the COMPORFN file are punched onto cards using a format of the first four (4) cards columns for the code number and the next sixteen (16) card columns for a description of the data items (Figure 2). Data items are punched one per card. The data cards are read into the program in ascending order of code numbers. These data cards are put into a one-dimensional array, each backward link element is initialized to zero and the array is written unformatted in the COMPORFN file. A card with 1001 punched in card columns one through four marks the end of the data for this file.

The input data for the locations and regions file, LOCRGDIR, are punched onto cards using a format of the first four card columns for the code and the next four card columns for an abbreviation of the location or region (Figure 3).
<table>
<thead>
<tr>
<th>Code</th>
<th>Abbreviation for competency, organization, or function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.** Input Card Format for Coding Competencies, Organizations, and Functions.
Figure 3. Input Card Format for Location Preference Codes.
The data cards are read into the program in ascending order of code numbers. These data are put into a one-dimensional array, each backward link element is set to zero and the array is written unformatted in the LOCRGDIR file. A card with 6001 punched in card columns one through four marks the end of the data for this file.

FILEDEVL Program

The FILEDEVL program is used to update the TOTALREC file, the INDEXREC file, the COMPORFN file, and the LOCRGDIR file. The updating of the TOTALREC file falls into four categories: "active" file establishment for students, file activation for alumni, "inactive" file establishment for students, and file inactivation. The updating of the COMPORFN file and LOCRGDIR file consists of initializing their respective backward link arrays, changing pointers in the arrays as appropriate for the TOTALREC update being done, and rewriting the arrays to the data sets. The INDEXREC update involves the addition of alumni social security numbers when appropriate.

The "active" file establishment for students procedure uses two (2) data cards punched from the Educational Placement Registration Activation Form. The input card formats are shown in App. D. In this updating procedure the TOTALREC record that is to be activated is retrieved, the backward link pointers for the previous records to use the same competency, teaching level and location preference backward links are set equal to this record number, the file is marked as an active file, and it is rewritten in the TOTALREC file. The saved links for the previous records are then used in a procedure that retrieves each record specified, sets its forward link equal to the record number of the activated record, and rewrites the record.
in the TOTALREC file. A message is then listed by the printer specifying that the record for the individual has been activated.

The file activation for alumni procedure uses three (3) input data cards punched from the Educational Placement Registration Form. The card formats shown in App. E. This procedure uses a subroutine called ADD which puts an alumnus' social security number in the proper position of the INDEX-REC file and returns a vacant record location in the TOTALREC file. The procedure then uses the same technique of setting backward and forward links as the student file activation procedure. The alumnus record is subsequently written in the vacant record position in the TOTALREC file.

The inactive file establishment procedure establishes a file for the student who wishes to register, but is not actively seeking employment at the time. The procedure uses one (1) input data card punched from the Registration Form. The card format is shown in App. C.

This file activation process involves retrieving the student's record from the TOTALREC file, marking it as a registered-inactive file, setting a code indicating his immediate plans, and rewriting the record in the TOTALREC file. No backward links or forward links are set for this type of file activation.

The file inactivation procedure uses one (1) input data card punched from the registration form. The card format is shown in App. B.

The file inactivation process involves retrieving the student's or alumnus' record from the TOTALREC file, saving the backward and forward links in the record, marking the file inactive, and setting a code indicating his immediate plans.

The record is then rewritten in the TOTALREC file. The procedure removes the inactivated record from the backward and forward link chains by
setting the forward link of its backward link equal to its forward link and
by setting the backward link of its forward link equal to its backward link.
Thus, the chain of active records is maintained leaving the inactivated
record intact in the master file for further data processing.

CONCLUSIONS

The basic design of the information system was used to expedite the
retrieval of information by the use of linked variables. This eliminates a
sequential search through all the records in the master file in order to
retrieve all the individuals interested in a particular teaching field or
location. This type of retrieval involves simply traversing the backward
link chain starting from the last record number that used the data item of
interest.

Another important consideration for utilizing this file structure is
the ease with which new records can be inserted and old records can be
deleted from the active file.

The information system provides the Educational Placement Section with
the means of maintaining a data base of registrant information on a direct
access device in compact data files. Retrieval of the data may be accom-
plished using any higher level programming language having an unformatted
input/output feature.
BIBLIOGRAPHY


Appendix A.

TOTALREC Record Variable List

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>MODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RWSSN</td>
<td>I4</td>
<td>Social Security Number</td>
</tr>
<tr>
<td>RNNAME(5)</td>
<td>5A4</td>
<td>Registrant's Name</td>
</tr>
<tr>
<td>DOB(3)</td>
<td>3I2</td>
<td>Registrant's date of birth</td>
</tr>
<tr>
<td>NUMCUR</td>
<td>I2</td>
<td>Numeric code for Registrant's curriculum (A &amp; R code)</td>
</tr>
<tr>
<td>RWALCU</td>
<td>A4</td>
<td>Alpha code for Registrant's curriculum (A &amp; R code)</td>
</tr>
<tr>
<td>SCHOOL</td>
<td>I2</td>
<td>Numeric code for Registrant's school (A &amp; R code)</td>
</tr>
<tr>
<td>CLASS</td>
<td>I2</td>
<td>Numeric code for Registrant's class (A &amp; R code)</td>
</tr>
<tr>
<td>RESCOD</td>
<td>I2</td>
<td>Numeric code for Registrant's residence status (A &amp; R code)</td>
</tr>
<tr>
<td>PLTEA</td>
<td>I2</td>
<td>Numeric code indicating Registrant's plans to teach 1 - yes, 2 - no, 3 - undecided</td>
</tr>
<tr>
<td>PLGRAD</td>
<td>I2</td>
<td>Numeric code indicating Registrant's plans to attend graduate school, 1 - yes, 2 - no, 3 - undecided</td>
</tr>
<tr>
<td>REGPC</td>
<td>I2</td>
<td>Numeric code indicating that Registrant has registered at the Placement Center, 1 - yes, 2 - no</td>
</tr>
<tr>
<td>RWZIP</td>
<td>I4</td>
<td>Registrant's permanent address, zip code</td>
</tr>
<tr>
<td>DATGRA(2)</td>
<td>2I2</td>
<td>Registrant's anticipated graduation date</td>
</tr>
<tr>
<td>WRCIT</td>
<td>A4</td>
<td>Registrant's citizenship</td>
</tr>
<tr>
<td>VET</td>
<td>I2</td>
<td>Registrant's veteran status (A &amp; R code)</td>
</tr>
<tr>
<td>SEXMAR</td>
<td>I2</td>
<td>Registrant's sex and marital code (A &amp; R code)</td>
</tr>
<tr>
<td>RWLPHO(2)</td>
<td>2A4</td>
<td>Registrant's local phone number</td>
</tr>
<tr>
<td>RWLADR(5)</td>
<td>5A4</td>
<td>Registrant's local address (street or box number)</td>
</tr>
<tr>
<td>RWLCS(5)</td>
<td>5A4</td>
<td>Registrant's local address (city/state)</td>
</tr>
<tr>
<td>RWLZIP</td>
<td>I4</td>
<td>Registrant's local zip code</td>
</tr>
<tr>
<td>RWPAD(5)</td>
<td>5A4</td>
<td>Registrant's permanent address (street or box number)</td>
</tr>
<tr>
<td>Field</td>
<td>Code</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>RWPSCS(5)</td>
<td>5A4</td>
<td>Registrant's permanent address (city/state)</td>
</tr>
<tr>
<td>RWPHON(2)</td>
<td>2A4</td>
<td>Registrant's permanent phone number</td>
</tr>
<tr>
<td>TYPCTRG</td>
<td>I2</td>
<td>Registrant's type of Placement Center Registration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0 - none, 1 - Educational Placement, 2 - B &amp; I, 3 - Both)</td>
</tr>
<tr>
<td>FILCOD</td>
<td>I2</td>
<td>Type of file code (1 - actively seeking employment,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - alumus seeking employment, 3 - inactive file, 4 - registered-not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>actively seeking employment)</td>
</tr>
<tr>
<td>TEALV1(3)</td>
<td>3I2</td>
<td>First Teaching Level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALVI(1) - Level Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALVI(2) - Backward link to previous record with same level 1 code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALVI(3) - Forward link to next record with same level 1 code</td>
</tr>
<tr>
<td>TEALV2(3)</td>
<td>3I2</td>
<td>Second Teaching Level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALV2(1) - Level code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALV2(2) - Backward link (not presently used)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEALV2(3) - Forward link (not presently used)</td>
</tr>
<tr>
<td>MAJARA(3)</td>
<td>3I1</td>
<td>First Competency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAJARA(1) - Competency code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAJARA(2) - Backward link to previous record with same first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>competency code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAJARA(3) - Forward link to next record with same first competency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>code</td>
</tr>
<tr>
<td>SUPFD1(3)</td>
<td>3I2</td>
<td>Supporting competency one</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD(1) - Competency code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD(2) - Backward link (not presently used)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD(3) - Forward link (not presently used)</td>
</tr>
<tr>
<td>SUPFD2(3)</td>
<td>3I2</td>
<td>Supporting competency two</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD2(1) - Competency code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD2(2) - Backward link (not presently used)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD2(3) - Forward link (not presently used)</td>
</tr>
<tr>
<td>SUPFD3(3)</td>
<td>3I2</td>
<td>Support competency three</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD3(1) - Competency code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD3(2) - Backward link (not presently used)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUPFD3(3) - Forward link (not presently used)</td>
</tr>
<tr>
<td>LOCPRI(3)</td>
<td>3I2</td>
<td>Location preference one</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCPRI(1) - Location preference code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCPRI(2) - Backward link to previous record with the same location</td>
</tr>
<tr>
<td></td>
<td></td>
<td>preference code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCPRI(3) - Forward link to next record with the same location</td>
</tr>
<tr>
<td></td>
<td></td>
<td>preference code</td>
</tr>
<tr>
<td>Field</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>LOCPR2(3)</td>
<td>3I2</td>
<td>Location preference two</td>
</tr>
<tr>
<td>LOCPR2(1)</td>
<td>- Location preference code</td>
<td></td>
</tr>
<tr>
<td>LOCPR2(2)</td>
<td>- Backward link (not presently used)</td>
<td></td>
</tr>
<tr>
<td>LOCPR2(3)</td>
<td>- Forward link (not presently used)</td>
<td></td>
</tr>
<tr>
<td>LOCPR3(3)</td>
<td>3I2</td>
<td>Location preference three</td>
</tr>
<tr>
<td>LOCPR3(1)</td>
<td>- Location preference code</td>
<td></td>
</tr>
<tr>
<td>LOCPR3(2)</td>
<td>- Backward link (not presently used)</td>
<td></td>
</tr>
<tr>
<td>LOCPR3(3)</td>
<td>- Forward link (not presently used)</td>
<td></td>
</tr>
<tr>
<td>LOPRCD</td>
<td>I2</td>
<td>Location preference code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)</td>
</tr>
<tr>
<td>MUSTCS</td>
<td>4A4</td>
<td>City/state for &quot;1&quot; location preference</td>
</tr>
<tr>
<td>AVAIL</td>
<td>I2</td>
<td>Date available code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1 - December, 2 - Jan-Feb, 3 - Jun-July, 4 - Aug-Sept)</td>
</tr>
<tr>
<td>HWCOMB</td>
<td>I2</td>
<td>Husband-Wife teaching combination code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1 - yes, 2 - no)</td>
</tr>
<tr>
<td>MINSAL</td>
<td>I2</td>
<td>Minimum salary acceptable - $100's</td>
</tr>
<tr>
<td>DGYR</td>
<td>I2</td>
<td>Degree and year awarded code (highest award)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1__ Bachelor's, 2__ Master's, 3__ Master's + 30, 4__ Doctorate's)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blank space will contain the last two digit of the year the degree was awarded)</td>
</tr>
<tr>
<td>YREXPR(3)</td>
<td>3I2</td>
<td>Years of employment experience</td>
</tr>
<tr>
<td>YREXPR(1)</td>
<td></td>
<td>- Years of teaching experience</td>
</tr>
<tr>
<td>YREXPR(2)</td>
<td></td>
<td>- Years of education administration experience</td>
</tr>
<tr>
<td>YREXPR(3)</td>
<td></td>
<td>- Years of non-education experience</td>
</tr>
<tr>
<td>POCRPL</td>
<td>I2</td>
<td>Post Graduation Plans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Code 0 - Code 7 --Diploma Receipt, Code 8 - seeking non-education employment)</td>
</tr>
<tr>
<td>INEMPL(4)</td>
<td>4A4</td>
<td>Employer (from diploma receipt, inactive card, or registered-inactive card)</td>
</tr>
<tr>
<td>INPOSI(3)</td>
<td>3A4</td>
<td>Position (from diploma receipt, inactive card, or registered-inactive card)</td>
</tr>
<tr>
<td>INADDR(5)</td>
<td>5A4</td>
<td>Address of employer (from diploma receipt, inactive card, or registered-inactive card)</td>
</tr>
<tr>
<td>INSALY</td>
<td>I4</td>
<td>Starting Salary (from diploma receipt, inactive card, or registered-inactive card)</td>
</tr>
<tr>
<td>Field</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INMOSR</td>
<td>I2</td>
<td>Months of service for which the starting salary is received</td>
</tr>
<tr>
<td>EDEMP (3)</td>
<td>312</td>
<td>Educational Employment Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDEMP (1) - Level code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDEMP (2) - Years of education experience code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDEMP (3) - Return to former position code</td>
</tr>
</tbody>
</table>
Appendix B.

Inactivation Input Card Format for FILEDEVL Program

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - INAC</td>
</tr>
<tr>
<td>5 - 13</td>
<td>I4</td>
<td>INSSN</td>
<td>Social Security Number</td>
</tr>
<tr>
<td>14</td>
<td>I4</td>
<td>INCODE</td>
<td>Post Graduation Plans Code (From KSU Diploma Receipt)</td>
</tr>
<tr>
<td>15 - 19</td>
<td>I4</td>
<td>INSALY</td>
<td>Salary for Job Accepted</td>
</tr>
<tr>
<td>20 - 21</td>
<td>I2</td>
<td>INMOSR</td>
<td>Months</td>
</tr>
<tr>
<td>22 - 37</td>
<td>4A4</td>
<td>INEMPL</td>
<td>Employer</td>
</tr>
<tr>
<td>38 - 49</td>
<td>3A4</td>
<td>INPOSI</td>
<td>Position Accepted</td>
</tr>
<tr>
<td>50 - 69</td>
<td>5A4</td>
<td>INADDR</td>
<td>City/State of Employment</td>
</tr>
</tbody>
</table>
Appendix C.

Registered - Inactive Input Card format for FILEDEVL Program

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHHR</td>
<td>Card Identifier - RIN4</td>
</tr>
<tr>
<td>5 - 13</td>
<td>I4</td>
<td>INSSN</td>
<td>Social Security Number</td>
</tr>
<tr>
<td>14</td>
<td>I4</td>
<td>INCODE</td>
<td>Post Graduation Plans Code (From KSU Diploma Receipt)</td>
</tr>
<tr>
<td>15 - 19</td>
<td>I4</td>
<td>INSALY</td>
<td>Salary for Job Accepted</td>
</tr>
<tr>
<td>20 - 21</td>
<td>I2</td>
<td>INMOSR</td>
<td>Months of Service for Salary</td>
</tr>
<tr>
<td>22 - 37</td>
<td>4A4</td>
<td>INEMPL</td>
<td>Employer</td>
</tr>
<tr>
<td>38 - 49</td>
<td>3A4</td>
<td>INPOSI</td>
<td>Position Accepted</td>
</tr>
<tr>
<td>50 - 69</td>
<td>5A4</td>
<td>INADOR</td>
<td>City/State of Employment</td>
</tr>
</tbody>
</table>
Appendix D.

File Activation Input Card Formats for FILEDEVL Program.
(Each file activation requires the two card set as described)

Card 1.

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - ACTV</td>
</tr>
<tr>
<td>5 - 24</td>
<td>5A4</td>
<td>INAME</td>
<td>Registrant's Name</td>
</tr>
<tr>
<td>25 - 33</td>
<td>I4</td>
<td>INSSN</td>
<td>Registrant's Social Security Number</td>
</tr>
<tr>
<td>34</td>
<td>I2</td>
<td>INSEX</td>
<td>Sex of Registrant</td>
</tr>
<tr>
<td>35 - 40</td>
<td>312</td>
<td>INDOB</td>
<td>Date of Birth</td>
</tr>
<tr>
<td>41 - 60</td>
<td>5A4</td>
<td>INADDR</td>
<td>Local Street Address</td>
</tr>
<tr>
<td>61 - 80</td>
<td>5A4</td>
<td>INCTYS</td>
<td>Local City/State</td>
</tr>
</tbody>
</table>

Card 2.

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - ACTV</td>
</tr>
<tr>
<td>5 - 9</td>
<td>I4</td>
<td>INZIP</td>
<td>Local ZIP Code</td>
</tr>
<tr>
<td>10 - 17</td>
<td>2A4</td>
<td>INLPHN</td>
<td>Local Phone Number</td>
</tr>
<tr>
<td>18</td>
<td>I2</td>
<td>INMAR</td>
<td>Marital Status</td>
</tr>
<tr>
<td>19 - 21</td>
<td>I2</td>
<td>INGYR</td>
<td>Degree &amp; Year Awarded (See DGYR description, App. A)</td>
</tr>
<tr>
<td>22 - 27</td>
<td>312</td>
<td>INYREX</td>
<td>Employment Experience (See YREXPR description, App. A)</td>
</tr>
<tr>
<td>28</td>
<td>I2</td>
<td>INHSWF</td>
<td>Husband-Wife Teaching Combination Code (1 - yes, 2 - no)</td>
</tr>
<tr>
<td>29 - 31</td>
<td>I2</td>
<td>INMINS</td>
<td>Minimum Salary Acceptable - $100's</td>
</tr>
<tr>
<td>32</td>
<td>I2</td>
<td>INAVIL</td>
<td>Date Available Code (1 - Dec, 2 - Jan-Feb, 3 - Jun-Jul, 4 - Aug-Sep)</td>
</tr>
<tr>
<td>Card Column</td>
<td>FORTRAN Node</td>
<td>Program Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>33 - 44</td>
<td>4I2</td>
<td>INCOMP</td>
<td>Teaching Competencies</td>
</tr>
<tr>
<td>45 - 50</td>
<td>2I2</td>
<td>INSTYP</td>
<td>Institution Type</td>
</tr>
<tr>
<td>51 - 62</td>
<td>3I2</td>
<td>INLOCP</td>
<td>Location Preferences Codes</td>
</tr>
</tbody>
</table>
| 63          | I2           | INLPRE           | Location Preference  
(0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference) |
| 64 - 79     | 4A4          | MUSTCS           | City/State for "a necessity"  
Location Preference |
| 80          | I4           | ICDTP            | Card Sequence Number - "2" |
Appendix E.

Alumnus File Activation Input Card Formats for FILEDEVL Program.
(Each alumnus file activation requires the three card set as described).

Card 1.

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - ALUM</td>
</tr>
<tr>
<td>5 - 24</td>
<td>5A4</td>
<td>INAME</td>
<td>Registrant's Name</td>
</tr>
<tr>
<td>25 - 33</td>
<td>I4</td>
<td>INSSN</td>
<td>Registrant's Social Security Number</td>
</tr>
<tr>
<td>34</td>
<td>I2</td>
<td>INSEX</td>
<td>Sex of Registrant</td>
</tr>
<tr>
<td>35 - 40</td>
<td>3I2</td>
<td>INDOB</td>
<td>Date of Birth</td>
</tr>
<tr>
<td>41 - 60</td>
<td>5A4</td>
<td>INADDR</td>
<td>Local Street Address</td>
</tr>
<tr>
<td>61 - 80</td>
<td>5A4</td>
<td>INCTYS</td>
<td>Local City/State</td>
</tr>
</tbody>
</table>

Card 2.

<table>
<thead>
<tr>
<th>Card Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - ALUM</td>
</tr>
<tr>
<td>5 - 9</td>
<td>I4</td>
<td>INZIP</td>
<td>Local ZIP Code</td>
</tr>
<tr>
<td>10 - 17</td>
<td>2A4</td>
<td>INLPHN</td>
<td>Local Phone Number</td>
</tr>
<tr>
<td>18 - 37</td>
<td>5A4</td>
<td>RWPAAD</td>
<td>Permanent Address (street or by number)</td>
</tr>
<tr>
<td>38 - 57</td>
<td>5A4</td>
<td>RWPCS</td>
<td>Permanent Address (city/state)</td>
</tr>
<tr>
<td>58 - 62</td>
<td>I4</td>
<td>RWPZIP</td>
<td>Permanent Address ZIP Code</td>
</tr>
<tr>
<td>63 - 70</td>
<td>2A4</td>
<td>RWPHON</td>
<td>Permanent Phone Number</td>
</tr>
<tr>
<td>71</td>
<td>I2</td>
<td>INMAR</td>
<td>Marital Status</td>
</tr>
<tr>
<td>72 - 74</td>
<td>I2</td>
<td>INDGYR</td>
<td>Degree and Year Awarded (See DGYR description, App. A)</td>
</tr>
</tbody>
</table>
### Appendix E (Continued)

<table>
<thead>
<tr>
<th>Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>75 - 79</td>
<td>-</td>
<td>-</td>
<td>Spaces</td>
</tr>
<tr>
<td>80</td>
<td>I4</td>
<td>ICDTP</td>
<td>Card Sequence Number - &quot;2&quot;</td>
</tr>
</tbody>
</table>

### Card 3.

<table>
<thead>
<tr>
<th>Column</th>
<th>FORTRAN Mode</th>
<th>Program Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 4</td>
<td>A4</td>
<td>INCHAR</td>
<td>Card Identifier - ALUM</td>
</tr>
<tr>
<td>5 - 10</td>
<td>3I2</td>
<td>INYREX</td>
<td>Employment Experience (See YREXPR description, App. A)</td>
</tr>
<tr>
<td>11</td>
<td>I2</td>
<td>INHSWF</td>
<td>Husband-Wife Teaching Combination Code (1 - yes, 2 - no)</td>
</tr>
<tr>
<td>12 - 14</td>
<td>I2</td>
<td>INMINS</td>
<td>Minimum Salary Acceptable - $100's</td>
</tr>
<tr>
<td>15</td>
<td>I2</td>
<td>INAVIL</td>
<td>Date Available Code (See AVAIL description, App. A)</td>
</tr>
<tr>
<td>16 - 27</td>
<td>4I2</td>
<td>INCOMP</td>
<td>Teaching Competencies</td>
</tr>
<tr>
<td>28 - 33</td>
<td>2I2</td>
<td>INSTYP</td>
<td>Institution Type</td>
</tr>
<tr>
<td>34 - 45</td>
<td>3I2</td>
<td>INLDCP</td>
<td>Location Preferences Codes</td>
</tr>
<tr>
<td>46</td>
<td>I2</td>
<td>INLPRE</td>
<td>Location Preference (0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)</td>
</tr>
<tr>
<td>47 - 62</td>
<td>4A4</td>
<td>MUSTCS</td>
<td>City/State for &quot;a necessity&quot; Location preference</td>
</tr>
<tr>
<td>63 - 79</td>
<td>--</td>
<td>--</td>
<td>Spaces</td>
</tr>
<tr>
<td>80</td>
<td>I4</td>
<td>ICDTP</td>
<td>Card Sequence Number - &quot;3&quot;</td>
</tr>
</tbody>
</table>
Appendix F.

Listing of FILEDEVL Program.
EXEC FORTCGLG PARM=PRT='MAP'

FILE=PROG FOR EDUCATIONAL PLACEMENT DATA FILES.

BY B. E. LUNSDING, AUGUST 1970. THIS PROGRAM ACCEPTS

THE FOLLOWING TYPES OF DATA CARDS IN ANY ORDER:

1. ACTIVATION CARDS  -- ACTV IN CC 1-4
2. ALUMNI CARDS   -- ALUM IN CC 1-4
3. INACTIVATION CARDS  -- INAC IN CC 1-4
4. REGISTERED  -- INACTIVE CARDS  -- IRAK IN CC 1-4

SEE USER MANUAL FOR COMPLETE DEFINITION OF THE ABOVE

DATA CARDS

MAIN ARRAYS USED BY THIS PROGRAM

INDEX(3400,2) -- THIS ARRAY CONTAINS THE SSN OF THE
INDIVIDUALS IN THE TOTALREC FILE AND A POINTER TO
THE LOCATION OF ALL THE INFORMATION PRESENTLY STORED
FOR THE INDIVIDUAL IN THE TOTALREC FILE

ICOMB(1000,8) -- THIS ARRAY CONTAINS THE BACKWARD LINK
INFORMATION FOR COMPETENCIES, TYPE OF ORGANIZATION AND
NON-TEACHING TITLES. BACKWARD LINKS ARE PROVIDED FOR UP TO
FOUR LEVELS OF COMPETENCIES AND UP TO TWO TEACHING AREAS

LOCAT(550,4) -- THIS ARRAY CONTAINS THE BACKWARD LINK
INFORMATION FOR LOCATION PREFERENCES. LINKS ARE PROVIDED
FOR UP TO THREE LOCATION PREFERENCES.

SEE APPENDIX F FOR A COMPLETE LISTING OF THE VARIABLES USED BY
THIS PROGRAM.

THIS PROGRAM USES THE REWRITE SUBROUTINE TO REFORMAT DATA AFTER IT
HAS BEEN INPUTTED UNDER CONTROL OF A DIFFERENT FORMAT.

SEE THE KSU COMPUTING CENTER NOTICE NUMBER 151 FOR ADDITIONAL
INFORMATION ABOUT THIS SUBROUTINE.

INTEGER*8 RNSSN, RNNAME(5), RWALCU, RWPZIP, RWGCT, RWLPH(2),
1RMLLR(5), RWLCS(3), RWZIP, RWPAD(5), RWPGS(5), RWPHUN(2),
2MUSTC(4), BUFI(41), TLBLK, TLRLK, MAJBKL, MAJFRL, LOCBLK,
3LOCFLK
INTEGER*2 DBB(3), NUMCUR, SCHOOL, CLASS, RECSCD, PLTFA, PLGHAD,
1RESPEC, DATGKA(12), VET, SEMP2, TELAVL(3), MAJARA(3),
2SUPP(3), SUPF(3), SPPF(3), LOPR(3), LOPR2(3), LOPR3(3),
3FILCOD, VAIL, INADEP, MNSAL, NGYR, NREPR(3), TYPGRC, LOPRCD, PGRPL,
4BUF2(18), INDV(3), INFOV, INVR, INNX, INXXS, INHS, INAV, INAV,
5INSE, DEEMPL(3), INMSR
DIMENSION INAME(5), INAUDR(5), INEMPL(4), INCTYS(5),
1INLPHN, INCGMP(4), INSTYP(2), INGCP(3), INPUS(3)
DIMENSION INPUT(80), ICOMB(1000,8), INOUT(200), LOCAT(550,4)
COMMON INDEX(3400,2), L1, L3
EQUIVALENCE (INPUT(1), INOUT(1))
DATA IACTV, IACTV, IRAK, IRKNA, IRAK, IRAK, IALUM, IALUM, IALUM, IALUM,
1INAC, INAC, INAC, INAC, INAC
DATA IBLNK, IBLNK, IBLNK, IBLNK, IBLNK, IBLNK, IBLNK, IBLNK, IBLNK
DEFINE FILE 30(40,800,L,K30), 31(11,800,L,K31)
DEFINE FILE 20(3400,320,L,K20), 21(8,3520,L,K21)
SET UP INDEX ARRAY FROM THE DATA SET

DO 10 J = 1,2
   DO 10 1 = 1,3400
10 INDEX(I,J) = 0
SET ALL THE ASSOCIATED VARIABLES TO ONE
K20 = 1
K31 = 1
K30 = 1
K21 = 1
L1 = 1
C READ A BLOCK OF RECORD FROM THE INDEXREC FILE.
C DEBLOCK INTO THE INDEX ARRAY
14 READ (21*K21) INPUT
   DO 15 I = 1,880,2
       J = 1 + 1
       IF (INPUT(I)) 20,20,16
   20 INDEX(L1,1) = INPUT(I)
       GO TO 22
   16 INDEX(L1,1) = INPUT(I)
       INDEX(L1,2) = INPUT(J)
   15 L1 = L1 + 1
       GO TO 14
22 WRITE(3,21) L1
21 FORMAT(' INDEX ARRAY SET UP WITH ',15,' RECORDS. ')
C END INDEX SET UP
C BEGIN SET UP OF COMPETENCIES, ORGAN. AND FUNCT., NON-TEACHING TITLES
   LEND = L1 - 1
   IFIXT = 0
   J25 = 25
   J25 = 1
C READ A BLOCK OF COMPETENCIES, ORGANIZATIONS, AND FUNCTIONS FROM
C THE COMP/ORGN FILE. DEBLOCK INTO THE ICOMP ARRAY.
   DO 32 I = 1,40
       READ(30*K30) INPUT
       ME = 0
       DO 33 J = J25,25
       33 ICOMP(J,K) = INPUT(ME)
       125 = 125 + 25
       J25 = J25 + 25
   32 END THIS SET UP
C BEGIN SET UP OF LOCATION PREFS. AND REGION PREFS.
   L50 = 1
   J50 = 50
   DO 42 I = 1,11
C READ A BLOCK OF LOCATIONS AND REGIONS FROM THE LOCGRDIR FILE.
C DEBLOCK INTO THE LOCAT ARRAY.
   READ(31*K31) INPUT
       ME = 0
       DO 43 J = L50,J50
       43 LOCAT(J,K) = INPUT(ME)
       150 = 150 + 50
       J50 = J50 + 50
   42 END THIS SET UP
C USE THE REREAD SUBROUTINE TO DETERMINE WHICH FORMAT TO USE
C CALL REREAD
5 FORMAT (44)
100 READ(1,5,END=1950) ICHAR
   IF (ICHAR .EQ. IACTV) GO TO 600
   IF (ICHAR .EQ. IRINA) GO TO 400
   IF (ICHAR .EQ. IALUM) GO TO 800
   IF (ICHAR .EQ. INAC) GO TO 200
C IF CONTROL PASSES TO THIS POINT, WRITE AN ERROR MESSAGE.
WRITE(3,101) ICHAR
101 FORMAT(' INPUT CARD ERROR. INVALID CARD TYPE *82X,A4)
GO TO 100
C
C BEGIN RECORD INACTIVATION FOR THE INDIVIDUAL
C
200 READ(199,201) INCHAR,INSSN,INCODE,INSALY,INMUSR,IVEMPL,INPOSI,
1 INADDR
C GO SEARCH FOR THE SSN IN THE INDEX ARRAY
CALL BINSCH(INSSN,IRECNU)
C IF NOT THERE WRITE ERROR MESSAGE
IF (IRECNU) 220,220,205
205 IF (IRECNU .GT. LEND) GO TO 208
GO TO 207
208 IFIXIT = -1
C STMTS 1200 TO 1214 + 1 CHECK FOR THE SSN IN THE DATA FILES
1200 WRITE(3,1199) INSSN
1199 FORMAT(' SEARCHING FOR ',',I10)
   L12 = L1-1
   DO 1201 MAX = 1,L12
      K20 = MAX
      READ(20*K20) L12SSN
      IF (L12SSN .EQ. INSSN) GO TO 1202
1201 CONTINUE:
   WRITE(3,1203) INSSN
1203 FORMAT(' RECORD FOR THIS NUMBER NOT IN TOTALREC FILE ',',I10)
   GO TO 100
1202 LREC = K20 -1
   DO 1210 MAX = 1,L1
      IF (INDEX(MAX,1).EQ. INSSN) GO TO 1212
9120 GO TO 1210
1212 LMAX = MAX
   GO TO 1214
1210 CONTINUE
   WRITE(3,1213) INSSN
1213 FORMAT(' SSN ',',IIC,' NOT IN INDEXREC FILE')
   GO TO 100
1214 INDEX(MAX,2) = LREC
   IRECN0 = LREC
   IF (IFIXIT) 207,207,418
207 K20 = IRECN0
   READ (20*K20) BUFI,TPRCG,FILCOD,TEAL1,TEAL2,MAJRA,SUPFD1,
1 SUPFD2,SUPFD3,LOGPR1,LOGPR2,LOGPR3,MUSTCS,AVAIL,HKWCOMB,INSAL,
2 DGYR,YREXPR,POGRPL
C CHECK FILCOD FOR AN ACTIVE FILE
IF (FILCOD .EQ. 1 .OR. FILCOD .EQ. 2) GO TO 260
C ELSE WRITE MESSAGE FOR INACTIVE FILE.
261 FORMAT(' THE FILE FOR THIS INDIVIDUAL IS NOT ACTIVE. INACTIVATION
1 TERMINATED. ',',IIC,3A4)
   WRITE(3,261) INSSN,BUFFI(2),BUFFI(3),BUFFI(4)
   GO TO 100
260 CONTINUE
C SAVE PART OF THE NAME
   INAME(1) = BUFFI(2)
   INAME(2) = BUFFI(3)
   INAME(3) = BUFFI(4)
C FILE INACTIVATION - CHECK FOR THIS RECORD BEING THE LAST RECORD
C IN THE LINK ARRAYS. IF IT IS SET THE LINK ARRAY FIELD EQUAL
C TO THE RESPECTIVE BACKWARD LINKS OF THE INACTIVATED RECORD.
IF (ICOMP(TEALV1(1),5) .EQ. IRECNO)
  1  ICOMP(TEALV1(1),5) = TEALV1(2)
IF (ICOMP(MAJARA(1),5) .EQ. IRECNO)
  1  ICOMP(MAJARA(1),5) = MAJARA(2)
ITEM1 = LOCPR1(1)
IF (ITEM1) 301,301,302
302 NITEM1 = ITEM1/100
  ITEM1 = ((ITEM1-11+10) + (ITEM1-(NITEM1*10)))
IF (LOCAT(IITEM1,2) .EQ. IRECNO)
  1  LOCAT(IITEM1,2) = LOCPR1(2)
301 CONTINUE

C SAVE THE BACKWARD AND FORWARD LINK VARIABLES

C TEACHING LEVEL ONE BACKWARD LINK
TLBLKLK=TEALV1(2)

C TEACHING LEVEL ONE FORWARD LINK
TLFRLK = TEALV1(3)

C MAJOR COMPETENCY BACKWARD LINK
MAJBLK = MAJARA(2)

C MAJOR COMPETENCY FORWARD LINK
MAJFRL = MAJARA(3)

C LOCATION PREFERENCES BACKWARD LINK
LOCBLK = LOCPR1(2)

C LOCATION PREFERENCES FORWARD LINK
LOCFRL = LOCPR1(3)

C SET POST GRADUATION PLANS CODE
POGRPL = INCODE

C SET DA TYPE CODE TO INACTIVE
FILCOD = 3

C SET DA POINTER TO RECORD AND WRITE UPDATED RECORD
K20 = IRECNO
WRITE(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1, TEALV2, MAJARA, SUPFD1,
  1  SUPFD2, SUPFD3, LOCPR1, LOCPR2, LOCPR3, MUSTCS, AVAIL, HWCOMB, MINSAL,
  2  OGRY, YKEXPR, POGRPL, INEMPL, INAPDR, INADAR, INSALY, INMOSR
C IF THIS BACKWARD LINK IS ZERO, BRANCH TO THE NEXT
IF (TLBLKLK) 230,230,231
231 K20 = TLBLKLK
READ(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1
C SET THE FORWARD LINK OF THE BACKWARD LINK EQ TO THE FORWARD
C LINK OF THIS RECORD
TEALV1(3) = TLFRLK
K20 = TLBLKLK
WRITE(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1
C IF THIS FORWARD LINK IS ZERO, BRANCH TO THE NEXT
230 IF (TLFRLK) 232,232,233
233 K20 = TLFRLK
READ(20*K20) BUFF1, TYPCRG, TEALV1

C SET THE BACKWARD LINK OF THIS RECORDS BACKWARD LINK EQUAL TO THE
C BACKWARD LINK OF THE INACTIVATED RECORD
TEALV1(21) = TLBLKLK
K20 = TLFRLK
WRITE(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1
C IF MAJOR AREA BACKWARD LINK IS ZERO, BRANCH TO THE NEXT
232 IF (MAJBLK) 234,234,235
235 K20 = MAJBLK
READ(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1, TEALV2, MAJARA
MAJARA(3) = MAJFRL
K20 = MAJBLK
WRITE(20*K20) BUFF1, TYPCRG, FILCOD, TEALV1, TEALV2, MAJARA
C IF MAJOR AREA FORWARD LINK IS ZERO, BRANCH TO THE NEXT
234 IF (MAJFRL) 236,236,237
237  K20 = MAJFLK
       READ(120,K20) BUFI1, TYPICRG, FICLDU1, TEALV1, TEALV2, MAJARA
       MAJARA(2) = MAJLKLK
       K20 = MAJFLK
       WRITE(20,K20) BUFI1, TYPICRG, FICLDU1, TEALV1, TEALV2, MAJARA
       IF LOCATION PRED BACKWARD LINK IS ZERO, BRANCH TO NEXT
236    IF (LDCBLK) 238, 238, 239
238    K20 = LDCBLK
       READ(120,K20) BUFI1, TYPICRG, FICLDU1, BUFI2, LDCPR1
       LDCPR1(3) = LDCFLK
       K20 = LDCBLK
       WRITE(20,K20) BUFI1, TYPICRG, FICLDU1, BUFI2, LDCPR1
       IF LOCATION PRED FORWARD LINK IS ZERO, BRANCH TO THE NEXT
238    IF (LDCFLK) 240, 240, 241
240    K20 = LDCFLK
       READ(120,K20) BUFI1, TYPICRG, FICLDU1, BUFI2, LDCPR1
       LDCPR1(2) = LDCBLK
       K20 = LDCFLK
       WRITE(20,K20) BUFI1, TYPICRG, FICLDU1, BUFI2, LDCPR1
240    WRITE(3,242) INSSN, INAME1, INAME2, INAME3
242    FORMAT(* INACTIVATION COMPLETE FOR NUMBER *, IN10, 1X, 3A4)
       GO TO 100

201    FORMAT(A4,19,11,15,12,4A4,3A4,5A4)
202    FORMAT(* INPUT ERROR IN CARD 'A4,19', NOT IN INDEX FILE)
220    WRITE(3,202) INCHAR, INSSN
       GO TO 100

C
C
END RECORD INACTIVATION
C
BEGIN REGISTERED - INACTIVE FILE PROCESSING
C
400    READ(199,201) INCHAR, INSSNA, INCODA, INSAVA, INMOSSR, INEMPLA,
       1 INPOS1, INADUR
C
GO PICK UP THE RECORD NUMBER
CALL BINSCH(INSSNA, IRECN1)
C
IF RECORD NOT FOUND OUTPUT ERROR MESSAGE
IF (IRECN1) 420, 420, 420
405    K20 = IRECN1
       READ (20,K20) BUFI1, TYPICRG, FICLDU1, TEALV1, TEALV2, MAJARA, SUPFD1,
       1 SUPFD2, SUPFD3, LDCPR1, LDCPR2, LDCPR3, MUSICS, AVALI, HWCOMB, MINSAL,
       2 DQYR, YREXPR, PDGPR1
       FILCUD = 4
       K20 = IRECN1
       WRITE(20,K20) BUFI1, TYPICRG, FICLDU1, TEALV1, TEALV2, MAJARA, SUPFD1,
       1 SUPFD2, SUPFD3, LDCPR1, LDCPR2, LDCPR3, MUSICS, AVALI, HWCOMB, MINSAL,
       2 DQYR, YREXPR, PDGPR1, INEMPLA, INPOS1, INADUR, INSAVA, INMOSSR
       WRITE(3,406) INSSNA, BUFI1(2), BUFI1(3), BUFI1(4)
        406 FORMAT(* REGISTERED - INACTIVE FILE ESTABLISHED FOR NUMBER *, 110,
       11X, 3A4)
       GO TO 100
420    WRITE(3,202) INCHAR, INSSNA
       GO TO 100

C C
C C
END REGISTERED - INACTIVE FILE PROCESSING
C C
BEGIN FILE ACTIVATION PROCESSING
C C
ACTIVATE THE FILE FOR THE INDIVIDUAL.
403 FORMAT (A4,15,2A4,11,13,312,11,13,11,613,314,11,4A4)
401 FORMAT (6A4,19,11,312,10A4)
402 FORMAT (A4,75X,1)
600 READ (99,401) ICCHAR, INAME, INSSN, INSEX, INDB, INADDR, INCTYS
C SET THE ALUMNUS FILE PROCESSING SWITCH OFF
IALSH = 0
C CHECK CARD CODE AND IF SECOND CARD
READ (1,402) ICHAR, ICOTP
IF (ICHAR .NE. IACTV) GO TO 410
IF (ICOTP .NE. 2) GO TO 410
READ (99,403) ICCHAR, INZIP, INLPHN, INMAR, INNGYR, INRREX, INHMSW, INMINS, INAVIL, INCMP, INSTYP, INLOCU, INLPRE, MUSTCS
C FIND THE TOTAL REC RECORD NUMBER FOR THIS SSN
CALL BINSCH(INSSN,IRECN0)
IF (IRECN0) 420,420,419
419 IF (IRECN0 .GT. LEAD) GO TO 500
GO TO 418
500 IFIXIT = 1
GO TO 1200
418 K20 = IRECN0
READ (20,K20) RWSNN, RNAME, DDB, NMCUR, RVALC, SCHOOL, CLAS, IRESCOD, PTEA, PLAGAD, REGC, RWZIP, DATGRA, RWCT, VET, SEXMAR, 2RWLPHN, RWAAD, RWALES, RWEA, RWPAD, RWPAP, RPMPHEN, TYPORC, FICOD, 3TEALV1, TEALV2, MAJARA, SUPFD1, SUPFD2, SUPFD3, LOCPRL1, LOCPRL2, 4LOCPR3, LOPRC
455 CONTINUE
C SET THE BACKLINK IN THIS RECORD FOR TEACHLEVEL ONE, MAJOR AREA
C AND LOCATION PREFERENCE ONE
ITEM1 = INCOMP(1)
MAJARA(2) = ICOMP(ITEMP1,5)
ICOMP(ITEMP1,5) = IRECN0
C
ITEM1 = INSTYP(1)
TEALV1(2) = ICOMP(ITEMP1,5)
ICOMP(ITEMP1,5) = IRECN0
C
C DECODE THE LOCATION PREFERENCE FOR LOCAT ARRAY
ITEM1 = INLOC(1)
NTEMP1 = ITEM1/100
ITEM1 = (ITEM1-1)*10+(ITEM1-NTEMP1*100)
LOCPRL1(2) = LOCAT(ITEM1,2)
LOCA(ITEM1,2) = IRECN0
C THIS RECORD NUMBER HAS NOW BEEN ENTERED INTO THE LINK ARRAYS AS
C THE LAST NUMBER TO USE THE PARTICULAR ITEM
C SAVE THE RECORD NUMBERS OF THE RECORDS WHICH MUST HAVE THEIR
C FRONT LINKS CHANGED
C
MAJARA
ITEM1 = MAJARA(2)
TEACHING LEVEL
NTEMP1 = TEALV1(2)
LOCATION PREFERENCE ONE
NTEMP1 = LOCPR1(2)
C
SET UP THE UPDATED INFO FOR WRITE
C TEACHING LEVELS ONE AND TWO CODES
TEALV1(1) = INSTYP(1)
TEALV2(1) = INSTYP(1)
C
MAJOR AREA CODE
MAJARA(1) = INCOMPL(1)

C SUPPORTING FIELDS ONE, TWO AND THREE CODES
SUPFD1(1) = INCOMPL(2)
SUPFD2(1) = INCOMPL(3)
SUPFD3(1) = INCOMPL(4)

C LOCATION PREFERENCE ONE, TWO, AND THREE CODES
LOCPR1(1) = INLOCPI(1)
LOCPR2(1) = INLOCPI(2)
LOCPR3(1) = INLOCPI(3)

C CHECK FOR UPDATED UDB
IF (INDUB(1) .EQ. 0 .OR. INDUB(2) .EQ. 0 .OR. INDUB(3) .EQ. 0)
1 GO TO 424

C UPDATE THE DATE OF BIRTH
DO 425 JX1=1,3
425 DOBH(JX1) = INDUB(JX1)
424 CONTINUE

C ZERO THE FRONT LINKS
TEALV1(3) = 0
TEALV2(3) = 0
MAJARA(3) = 0
LOCPR1(3) = 0
SUPFD1(3) = 0
SUPFD2(3) = 0
SUPFD3(3) = 0
LOCPR2(3) = 0
LOCPR3(3) = 0

C SET THE CODE FOR THE LOCATION PREFERENCE
LORCDE = INLPRE

C CHECK FOR PROCESSING ALUMNI CARDS
IF (IALSWH) 426, 426, 865

426 CONTINUE

C WRITE THE ACTIVATED RECORD IN THE TOTALREC FILE
C SET PC REG TO ED PLAC AND THE FILE TYPE TO ACTIVE
TYPERC = 1
FILCOD = 1

475 K20 = IRECNO
WRITE(20*K20) RWSSN, INAME, DOB, NUMCUR, RWALCU, SCHOOL, CLASS, RESCOD,
     IPLTE, PLGRAD, REGRC, RPZIP, DATGRA, RWGIT, VET, SEXMAR, INLPHN, INADDR,
     2INCYS, 2INZP, 2WPAC, 2WPC5, 2RPHON, 2YPCRG, 2FILCOD, 2TEALV1, 2TEALV2,
     3MAJARA, 3SUPFD1, 3SUPFD2, 3SUPFD3, 3LOCPR1, 3LOCPR2, 3LOCPR3, 3LORCDE, 3MUSTCS,
     4INAV, 4INHISW, 4INMNS, 4INGYR, 4INREX

C CHANGE THE FRONT LINKS IN THE BACK LINKS

C TEACHING LEVEL
IF (INETEMP) 430, 430, 431

431 K20 = NETEMP1
READ(20*K20) BUFFI, TYPERC, FILCOD, TEALV1
TEALV1(3) = IRECNC
K20 = NETEMP1
WRITE(20*K20) BUFFI, TYPERC, FILCOD, TEALV1

C MAJOR AREA
430 IF (INETEMP) 432, 432, 433

433 K20 = ITEMP1
READ(20*K20) BUFFI, TYPERC, FILCOD, TEALV1, TEALV2, MAJARA
MAJARA(3) = IRECNC
K20 = ITEMP1
WRITE(20*K20) BUFFI, TYPERC, FILCOD, TEALV1, TEALV2, MAJARA

C LOCATION PREFERENCE
432 IF (INETEMP) 440, 440, 435

435 K20 = NETEMP1
READ(20*K20) BUFFI, TYPERC, FILCOD, BUF2, LOCPR1
LOCPR1(3) = IRECNC
K2C = MTEMP1
WRITE(20',K2C)BUFF1,TYPCHR,FILCOD,BUF2,LOCPR1

C CHECK FOR PROCESSING ALUMNUS CARDS
440 IF (IALSHW) 445,465,870
445 WRITE(3,446)RNSSN,RWNAME(1),RWNAME(2),RWNAME(3)
446 FORMAT('FILE ACTIVATED FOR NUMBER ',I10,1X,3A4)
GO TO 100
410 WRITE(3,411)INAME,INSSN
411 FORMAT('INVALID INPUT RECORD FOR ',5A4,I10)
GO TO 100

C END FILE ACTIVATION
C
C BEGIN ACTIVATION OF ALUMNUS FILE
C
801 FORMAT(6A4,19,11,112,10A4)
805 FORMAT(A4,11,11,11,613,314,11,4A4)
803 FORMAT(A4,15,12A4,15,2A4,11,11)
C SET THE ALUMNUS FILE PROCESSING SWITCH TO ON (1)
800 IALSHW = 1
READ(99,801)INCHAR,INAME,INSSN,INSEX,INDOB,INADDR,INCTYS
C CHECK CARD CODE AND IF SECOND CARD
READ(1,402)ICHR,ICDTDP
IF (ICHR.NE.IALUM) GO TO 810
IF (ICDTDP.NE.2) GO TO 810
READ(99,803)INCHAR,INZIP,INLPHN,RWPAD,RWPICS,RWPZIP,RWPHON,INMAR,
11IPGYR
C CHECK CARD CODE AND IF THIRD CARD
READ(1,402)ICHR,ICDTDP
IF (ICHR.NE.IALUM) GO TO 811
IF (ICDTDP.NE.3) GO TO 811
READ(99,805)INCHAR,INREX,INHSWF,INMINS,INAVIL,INCOMB,INSTYP,
11INOCIP,INLPRE,MUSTCS
C GO FIND THE TOTALREC RECORD NUMBER FOR THIS SSN
CALL BINSCH(INSSN,IRECNO)
C IF RECORD NUMBER FOUND GO WRITE ERROR MESSAGE
IF (IRECNO) 850,850,860
860 WRITE(3,861)INAME,INSSN
861 FORMAT(1X,5A4,I10,'--THIS RECORD HAS ALREADY BEEN ENTERED OR THE
11INSSN IS IN ERROR!)
GO TO 100
810 WRITE(3,411)INAME,INSSN
READ(1,402)ICHR,ICDTDP
GO TO 100
811 WRITE(3,411)INAME,INSSN
GO TO 100
C GO AND THIS RECORD TO THE INDEX ARRAY AND RETURN ITS RECORD NUMBER
850 CALL ADD(INSSN,IRECNO)
C SET PC REG TO ED. PLACE. AND THE FILE TYPE TO ALUMNUS
TYPCHR=1
FILCOD=2
C GO SET UP LINKS AS SPECIFIED
GO TO 455
865 CONTINUE
C SET UP THOSE VARIABLES NORMALLY TAKEN FROM THE ACR FILE
NUMCUR=0
RWALGCU=1BLNK
SCHOOL=0
CLASS=0
RESCO0=0
PLEA0=0
PLGRAD=0
RGCPC=0
DAGRA(1)=0
DAGRA(2)=0
RTCIT=1BLNK
VET=0
SSEXIN=INSEX

C GO WRITE THE RECORD AND SET THE LINKS
  GO TO 475
C
871 FORMAT('ALUMNUS FILE ACTIVATED FOR NUMBER 'I10,1X,3A4)
870 WRITE(3,871) INSN,INAME(1),INAME(2),INAME(3)
  GO TO 100
C END OF ALUMNUS RECORD PROCESSING
C
C STORE THE UPDATED INDEX ARRAY, ICOMP ARRAY, AND LOCAT ARRAY ON
C THEIR RESPECTIVE DATA SETS
C
C RESTORE THE INDEX ARRAY BY BLOCKING AND WRITING ON THE DISK.
  1950 LI=1
      K30 = 1
      K31 = 1
      K21=1
  1995 DO 2000 I=1,880,2
     J=I+1
     IF(INDEX(1,I)) 2010,2010,2005
  2010 INPUT(I) = INDEX(1,I)
     GO TO 2013
  2005 INPUT(I) = INDEX(1,I)
     INPUT(1) = INDEX(1,1)
     INPUT(3) = INDEX(1,2)
  2000 LI=LI+1
     WRITE(21,K21) INPUT
     GO TO 1995
  2015 WRITE(21,K21) INPUT
C RESTORE THE ICOMP ARRAY BY BLOCKING AND WRITING ON THE DISK.
  125 = 1
  J25 = 25
  DO 2032 I=1,40
     ME = 0
     DO 2021 LE=I25,J25
       DO 2021 KE=1,8
           ME=ME+1
     2021 INPUT(ME) = ICOMP(LE,KE)
       WRITE(30,K30) INPUT
     125 = 125+25
  2032 J25 = J25+25
C END ICOMP RESTORE
C RESTORE THE LOCAT ARRAY BY BLOCKING AND WRITING ON THE DISK.
  150 = 50
  I50 = 1
  DO 2042 I=1,11
     ME = 0
     DO 2043 J=I50,J50
       DO 2043 K=1,4
           ME = ME+1
     2043 INPUT(ME) = LOCAT(J,K)
       WRITE(31,K31) INPUT
     I50 = I50+50
2042 J50 = J50+50
2052 WRITE(*,2052)
2062 FORMAT(* NORMAL END OF JOB*)
       END
       SUBROUTINE BINSCH (ISSN, IRECN)
       C THIS SUBROUTINE SEARCHES THE INDEX ARRAY FOR THE GIVEN SOCIAL
       C SECURITY NUMBER. IF THE SSN IS IN THE ARRAY THE RECORD NUMBER
       C FOR THE SSN IN THE TOTALREC FILE IS RETURNED.
       C IF THE SSN IS NOT IN THE ARRAY, THE POSITION OF WHERE IT SHOULD
       C BE PLACED TO MAINTAIN THE ASCENDING ORDER OF THE INDEX ARRAY
       C IS RETURNED.
       COMMON INDEX(3400,2), L1, L3
       I = (L1 + 1) / 2
       INCR = (I + 1) / 2
       LEND = L1 - 1
       1 CONTINUE
       IF (ISSN .EQ. INDEX(I,1)) 2,3,4
       2 IF (INCR .EQ. 1) GO TO 15
       IF (INCR .LT. 1) GO TO 5
       I = I - INCR
       INCR = (INCR + 1) / 2
       IF (I .LT. 1) I = I + INCR
       GO TO 1
       3 IRECN = INDEX(I,2)
       RETURN
       15 I = I - INCR
       INCR = 0
       GO TO 1
       16 I = I + INCR
       INCR = 0
       GO TO 1
       5 J = I - 10
       IF (J .LT. 1) J = 1
       K = I + 10
       IF (K .GT. LEND) K = LEND
       DO 6 L = J,K
       IF (ISSN .NE. INDEX(L,1)) GO TO 6
       M = L
       GO TO 7
       6 CONTINUE
       IRECN = 0
       L3 = I
       RETURN
       7 IRECN = INDEX(M,2)
       RETURN
       END
       SUBROUTINE ADD (ISSN, IRTN)
       C THIS SUBROUTINE USES THE RELATIVE POSITION RETURNED FROM THE
       C BINSCH SUBROUTINE TO PLACE THE SSN IN ITS PROPER POSITION IN
       C THE INDEX ARRAY, THE POINTER TO THE NEXT AVAILABLE POSITION
       C IN THE TOTALREC FILE IS ASSIGNED TO THE NEW SSN ENTRY, AND
       C INCREMENTED TO POINT TO A NEW POSITION.
       COMMON INDEX(3400,2), L1, L3
IL1 = L1 + 1
IF (INDEX(L3,1).LT. ISSN) GO TO 20
IF (INDEX(L3,1).GT. ISSN) GO TO 50
IRTN = -1
RETURN
20 IL3 = L3 + 2
24 INDEX(LIL1,1) = INDEX(IL1-1,1)
INDEX(LIL1,2) = INDEX(IL1-1,2)
IF (L1.LT.L3) GO TO 25
IL1 = (IL1 -1)
GO TO 24
25 INDEX(IL3-1,1) = ISSN
INDEX(IL3-1,2) = L1+1
L1 = L1+1
IRTN = L1
RETURN
50 IL3 = L3 + 1
GO TO 24
END

//GO.SYSPRN UD SYSOUT=A
//TF20F001 DD DS=CDP12PC.TOTALREG.REL,UNIT=SYSDA, VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=320)
//FT21F001 DD DS=CDP12PC.INDEXREG.REL,UNIT=SYSDA, VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=3520)
//FT30F001 DD DS=CDP12PC.COMPORFN.REL,UNIT=SYSDA, VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//FT31F001 DD DS=CDP12PC.LOCORDIR.REL,UNIT=SYSDA, VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//GO.SYSIN DD *
INAC51348145 100001UAL  SYS ANAL DENVER, COLO
ACTYLUNASINGER,RICHARD  515348145112233980X 114  LUISVILLE,KANSAS
ACTY664504567298 2270 0 0 0 760 010 1601 2 2
RINA51346397710000G12KSU COMP CNT R SYS ANAL MANHATTAN,KANSAS
ALUMROBINETT,SHERRY D. 49552535720420489046 WALNUT ST. KANSAS CITY, MO
ALUM DE313832 9046 WALNUT KANSAS CITY, MO DE318321170 2
ALUM 0 0 02 04760 010 16020508 1KANSAS CITY,MO
INAC51348145 100001UAL  SYS ANAL DENVER, COLO
/
## Additional FILEDEVL Program Variables

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>MODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFP1(41)</td>
<td>41I4</td>
<td>Spacer variable - used to facilitate disk reads and writes</td>
</tr>
<tr>
<td>TLBKLK</td>
<td>I4</td>
<td>Temporary storage variable for teaching level one backward link</td>
</tr>
<tr>
<td>TLFRLK</td>
<td>I4</td>
<td>Temporary storage variable for teaching level one forward link</td>
</tr>
<tr>
<td>MAJBLK</td>
<td>I4</td>
<td>Temporary storage variable for major area backward link</td>
</tr>
<tr>
<td>MAJFRL</td>
<td>I4</td>
<td>Temporary storage variable for major area forward link</td>
</tr>
<tr>
<td>LOCBLK</td>
<td>I4</td>
<td>Temporary storage variable for location preference one backward link</td>
</tr>
<tr>
<td>LOCFLK</td>
<td>I4</td>
<td>Temporary storage variable for location preference one forward link</td>
</tr>
<tr>
<td>BUF2(18)</td>
<td>18I2</td>
<td>Spacer variable - used to facilitate disk read and writes</td>
</tr>
<tr>
<td>INDOB(3)</td>
<td>312</td>
<td>Card input variable - date of birth</td>
</tr>
<tr>
<td>INDCYR</td>
<td>I2</td>
<td>Card input variable - degree and year awarded code</td>
</tr>
<tr>
<td>INYREX(3)</td>
<td>312</td>
<td>Card input variable - years of experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INYREX(1) - years of teaching experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INYREX(2) - years of educational administrative experience</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INYREX(3) - years of non-educational experience</td>
</tr>
<tr>
<td>INHSWF</td>
<td>I2</td>
<td>Card input variable - husband-wife teaching combination (1 - yes, 2 - no)</td>
</tr>
<tr>
<td>INMINNS</td>
<td>I2</td>
<td>Card input variable - minimum salary acceptable---$100's</td>
</tr>
<tr>
<td>INSEX</td>
<td>I2</td>
<td>Card input variable - sex code (1 - male, 2 - female)</td>
</tr>
<tr>
<td>INCTYS(5)</td>
<td>5A4</td>
<td>Card input variable - registrant's local city and state</td>
</tr>
<tr>
<td>INLPDN(2)</td>
<td>2A4</td>
<td>Card input variable - registrant's local phone number</td>
</tr>
<tr>
<td>Variable</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INCOMP(4)</td>
<td>4I4</td>
<td>Card input variable - competencies</td>
</tr>
<tr>
<td>INSTYP(2)</td>
<td>2I4</td>
<td>Card input variable - institutions or organizations</td>
</tr>
<tr>
<td>INLOCP(3)</td>
<td>3I4</td>
<td>Card input variable - location preference codes</td>
</tr>
<tr>
<td>ISSN</td>
<td>I4</td>
<td>Card input variable - registrant's social security number</td>
</tr>
<tr>
<td>INCHAR</td>
<td>A4</td>
<td>Card input variable - card identification number</td>
</tr>
<tr>
<td>INCODE</td>
<td>I4</td>
<td>Card input variable - post graduation plans code</td>
</tr>
<tr>
<td>IRECNO</td>
<td>I4</td>
<td>Pointer to record number in TOTALREC file</td>
</tr>
<tr>
<td>INZIP</td>
<td>I4</td>
<td>Card input variable - registrant's local zip code</td>
</tr>
<tr>
<td>INMAR</td>
<td>I4</td>
<td>Card input variable - registrant's marital code</td>
</tr>
<tr>
<td>INLPRE</td>
<td>I4</td>
<td>Card input variable - location requirement (0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)</td>
</tr>
<tr>
<td>ITEMP1</td>
<td>I4</td>
<td>Temporary storage variable</td>
</tr>
<tr>
<td>NTEMP1</td>
<td>I4</td>
<td>Temporary storage variable</td>
</tr>
<tr>
<td>MTEMP1</td>
<td>I4</td>
<td>Temporary storage variable</td>
</tr>
</tbody>
</table>
Appendix G.

Listing of CREATE Program.
CREATED PROGRAM BY RICHARD LONSENGER, AUGUST 1970.

THE PROGRAM ACCEPTS ONE DATA CARD WHICH HAS THE MAXIMUM

GRADUATION DATE OF INTEREST PRINTED IN MTH/YEAR FORMAT IN CARD

COLUMNS 1-4. THE PROGRAM COMPARES THIS DATE WITH THE GRADUATION

DATE OF ALL STUDENTS ON THE ADMISSIONS AND RECORDS MASTER TAPE

RETRIEVING THOSE THAT LIE WITHIN THIS DATE. THE PERTINENT

INFORMATION IS FORMATTED AND WRITTEN IN THE TOTALREC FILE. THE

POSITION OF THE RECORD IN THE FILE IS SAVED ALONG WITH THE SOCIAL

SECURITY NUMBER OF THE STUDENT IN THE INDEXREC FILE. INDEXREC

FILE CONTAINS THE SOCIAL SECURITY NUMBERS IN ASCENDING ORDER.

CREATED CSECT
SAVE (14,12),* *
PRINT NDGEN
BALA 12,0
USING *,12
ST 13,SAVE+4
LA 13,SAVE
B BEGIN
SAVE DS 18F
BUFFIN DS CL480
BUFFOUT DS CL320
INPUT DS CL80
M1 DC F'F-1'
*
* READ A CARD THAT CONTAINS THE MAXIMUM GRADUATION DATE FOR THE FILE
* BEGIN OPEN (CARDIN,INPUT) OPEN CARD FILE
READ CARDEC,SR,CARDIN,INPUT,'S' READ MAX GRAD. DATE
CHECK CARDEC
CLOSE CARDIN CLOSE CARD FILE
LA 5,BIGREC GET THE ADDR OF THE INDEX FILE BUFFER
L 4,FOUR*40 LOAD CONSTANT OF 440
MVI BUFFOUT,X'00' ZERO OUT BUFFOUT
MVC BUFFOUT+1(256),BUFFOUT
MVC BUFFOUT+256(63),BUFFOUT
*
* OPEN THE TAPE FILE AND THE TWO DISK FILES
OPEN (TAPEIN,INPUT),DISKOUT,(OUTPUT),DISKFILE,(OUTPUT))
LA 6,6
LA 2,3399 SET THE TOTALREC FILE COUNTER
LA 11,8 SET THE INDEXREC FILE COUNTER
LA 3,1 SET UP THE INDEX POINTER
LA 10,BUFFIN=1
READ READ TAPEDEC,SR,TAPEIN,BUFFIN,'S' GET A RECORD
CHECK TAPEDEC SR CHECK FOR SUCCESSFUL READ
CLI 40B(10,X'40' CHECK FOR BLANK GRAD DATE
BE REED GO READ ANOTHER REC IF IT IS
*
* CHECK FOR VALID GRADUATION DATE
CLI 40B(10,X'FO'
BL REED
CLI 406(10,X'FO'
BL REED
*
* END GRAD DATE VALIDITY CHECK
CLC 407(2,10),INPUT+2 CHECK YEAR LT MAX
BH REED
CLC 405(2,10),INPUT CHECK FOR MONTH LT MAX
BH REED
B  KEEP
SPACE 3

* SET UP A RECORD FOR TOTALREC AND RASTERFILE
DS UD
TEMP DS D
COUNT DC F'10'
BLANKS DC X'44444440'
OF DC X'0000000C0F'
FCF040 DC F'1440'
MACLNCN DC X'0000'
FCF0C DC X'F000F000'
SPACE 3

KEEPE XC TEMP(0),TEMP ZERO TEMP
CLI 9(110),X'40' CHECK FOR BLANK SSN
BE READ DONT INCLUDE IF IT IS
CLI 9(1100),X'FO'
BL REC
CLI 1(10),X'FO'
BL REED
PACK TEMP(0),119(10) CHANGE SSN TO
CVB 9,TEMP AN I4 NUMBER
ST 9,BUFFOUT PUT SSN IN OUTPUT BUFFER
L B,COUNT SET UP REC NUMB FOR
LA 8,1(8) THE INDEX FILE
ST 8,COUNT SAVE THE COUNT
ST 9,BLANKS SAVE THE SSN
MVC 0(4,5),BLANKS
ST 3,4(5) SAVE THE REC NUMB FOR THIS SSN
LA 3,1(3) INCR POINTER
BCTR 2,0 REDUCE THE RECORD COUNT
MVC BUFFOUT+4(19),10(10) GET THE NAME

* SET UP DATE OF BIRTH
XC TEMP(0),TEMP
CLI 371(110),X'40' CHECK FOR BLANK BIRTH MONTH
DNE **10 OK
MVC 370(12,10),MACLNCN
PACK TEMP(0),370(12,10) CHANGE MONTH
CVB 9,TEMP TO BINARY
STH 9,BUFFOUT+24 STORE THE MONTH OF BIRTH
CLI 373(110),X'40' CHECK FOR BLANK BIRTHDAY
DNE **10
MVC 372(12,10),MACLNCN
PACK TEMP(0),372(12,10) GET THE DAY OF BIRTH
CVB 9,TEMP AND CONVERT TO BINARY
STH 9,BUFFOUT+26 STORE THE DAY OF BIRTH
CLI 90(10),X'40' CHECK FOR BLANK BIRTH YEAR
BNE **10
MVC 89(2,10),MACLNCN
PACK TEMP(8),69(2,10) GET THE YEAR OF BIRTH
CVB 9,TEMP
STH 9,BUFFOUT+28 STORE THE YEAR OF BIRTH
CLI 34(10),X'40' CHECK FOR BLANK NUMERIC Curr
MVC MACLNCN
CLI 34(10),X'FO'
BNE PACNC
MOVNC
DC 32(3,10),FCF0FO
PACNC PACK TEMP(8),32(3,10) GET THE NUMERIC CURRICULUM
CVB 9,TEMP
STH 9,BUFFOUT+30 STORE THE NUMERIC CURRICULUM
BUFFOUT+32(4),35(10) STORE THE ALPHA CURRICULUM

BUFFOUT+36(10) GET THE SCHOOL CODE

BUFFOUT+36(10) GET THE SCHOOL CODE

BUFFOUT+38(10) STORE THE CLASS CODE

BUFFOUT+38(10) STORE THE CLASS CODE

BUFFOUT+40(10) STORE RESIDENCE CODE

BUFFOUT+40(10) STORE RESIDENCE CODE

BUFFOUT+42(10) DOES PLAN TO TEACH EQUAL Y?

BUFFOUT+44(10) DOES PLAN GRAD SCHOOL EQUAL Y?

BUFFOUT+46(10) DOES REG AT PLACE CENTER EQ Y?

BUFFOUT+48(10) DOES REG AT PLACE CENTER EQ Y?

BUFFOUT+48(10) DOES REG AT PLACE CENTER EQ Y?

BUFFOUT+50(10) DOES REG AT PLACE CENTER EQ Y?

BUFFOUT+52(10) GET THE MONTH OF GRADUATION

BUFFOUT+54(10) GET THE YEAR OF GRADUATION

BUFFOUT+56(10) MOVE THE CITIZENSHIP

BUFFOUT+58(10) MOVE THE CITIZENSHIP

BUFFOUT+60(10) GET SEX/MAR CODE

BUFFOUT+62(10) GET LOCAL PHONE NUMB

BUFFOUT+64(10) GET LOCAL ADJUR

BUFFOUT+66(10) GET CITY/STATE
CLI 14C(10),X'40'  CHECK FOR BLANK ZIP CODE
BE NEX1ZIP
CLI 14C(10),X'FO'
BL NEX1ZIP
CLI 136(10),X'40'
BE NEX1ZIP
PACK TEMP(B)+136(5,10)  GET ZIP CODE
CVB 9,TEMP
ST $4,BUFFOUT+112
NEX1ZIP
CLI 344(10),X'40'  CHECK FOR BLANK ZIP
BE RITE
CLI 344(10),X'FO'
BL RITE
CLI 34C(10),X'FO'
BL RITE
PACK TEMP(B)+136(5,10)  GET ZIP CODE
CVB 9,TEMP
ST $4,UFFOUT+48  STORE PERM ADDR ZIP CODE
RITE
NOP *
MVC BUFFOUT+116(20),301(10)  GET PERMANENT ADDR
MVC BUFFOUT+136(18),322(10)  GET CITY/STATE
MVC BUFFOUT+166(7),348(10)  GET PERM RES PHONE NUMBER
SPACE 3
WRITE DFIVECB,SF,DISKFILE,BUFFOUT,'S'  WRITE TOTAL RECORD
SPACE 2
CHECK DFIDECEB  CHECK FOR SUCCESSFUL WRITE
LA 5,B(5)  INK THE POINTER IN BIGREC
CR $8,4  CHECK FOR 440 IN BIGREC
BL REED  GO READ ANOTHER TAPE RECORD
SPACE 2
WRITE DUOUTDECB,SF,DISKOUT,BIGREC,'S'
SPACE 2
CHECK DUOUTDEC
SPACE 2
BCTR 11,0  DCR BIGREC COUNT
SR $8,8
ST $8,COUNT  ZERO COUNT
LA 5,BIGREC
B REED
TAPEEND
MVC 04(5),M1  MOVE IN A -1
SR 9,9
STH 9,4(5)
WRITE DUOUTDEC,SF,DISKOUT,BIGREC,'S'
SPACE 2
CHECK DUOUTDEC
BCTR 11,0
* FORMAT THE REMAINDER OF TOTALREC AND INDEXREC
MVI BUFFOUT,X'00'  ZERO OUT BUFFOUT
MVC BUFFOUT+1256,BUFFOUT
MVC BUFFOUT+256(63),BUFFOUT
LCOP1
WRITE TOTUECB,SF,DISKFILE,BUFFOUT,'S'
CHECK TOTUECB
BCT 2,LCOP1
LCOP2
WRITE INXDECB,SF,DISKOUT,BIGREC,'S'
CHECK INXDECB
BCT 11,LCOP2
* END FORMATTING
SPACE 2
CLOSE TAPEIN,DISKFILE,DISKOUT
SPACE 2
L 13,SAVER+4
RETURN (14,12)
SPACE 2
CARDIN DCB ODNAME=SYSIN,DSORG=PS,RECFM=F,BLKSIZ=40,MACRF=R
TAPEIN DCB ODNAME=TAPEMST,DSORG=PS,RECFM=F,BLKSIZ=480,MACRF=R, X
DEV=TA,CEN=2,TRTCH=C,EUOD=TAPEND
DISKFILE DCB ODNAME=TCTALFIL,DSORG=PS,RECFM=F,BLKSIZ=320,MACRF=W
DISKOUT DCB ODNAME=DISKMAST,DSORG=PS,RECFM=F,BLKSIZ=3520,MACRF=W
BIGREC DS CL3520
END CREATED
/*
//GO.SYSDUMP DD SYSOUT=A
//GO.DISKMAST DD DSN=GC0P12PC.INDEXREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),SPACE=(TRK,(14)),DCB=(RECFM=F,BLKSIZ=3520)
//GO.TOTALFIL DD DSN=GC0P12PC.TOTALREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),SPACE=(TRK,(160,10)),
// DCB=(RECFM=F,BLKSIZ=320)
//GO.TAPEMST DD UNIT=TAPE7,DISP=(OLD,KEEP),LABEL=(2,NL),DSN=RELPC, X
// VOL=SER=MYTAPE
//GO.SYSIN DD *
C970
/*
Appendix H.

Listing of DIRCREA Program.
EXEC FORTRAN

SUBROUTINE PROGRAM BY RICHARD LIONSGER, AUGUST 1970.

THIS PROGRAM SETS UP THE TWO Files, CUMHOURS AND LADEGOIR, THAT
C WILL CONTAIN THE BACKWARD LINK POINTERS FOR COMPETENCIES, TEACHING
C LEVELS, NON-TEACHING FUNCTIONS, AND LOCATION PREFERENCES. THE
C MAIN ACTIVITY OF THIS PROGRAM IS TO BLOCK THE INPUT INTO AN 860
C BYTE ONE DIMENSIONAL ARRAY FOR UNFORMATTED OUTPUT ON THE FILES.
C COMPETENCIES, ORGANIZATIONS, AND FUNCTIONS MUST BE PUNCHED ONE PER
C CARD ACCORDING TO FORMAT ONE (1). THE END OF THIS GROUP OF DATA
C CARDS IS MARKED BY A 101 CARD (CC 1-4). THE LOCATION PREFERENCE
C DATA CARDS MUST BE PUNCHED ONE PER CARD ACCORDING TO FORMAT TWO (2).
C THE END OF THIS GROUP OF CARDS IS MARKED BY A 601 CARD (CC 1-4).
C C IN THIS GROUP THE FIRST TWO DIGITS ARE CONSIDERED TO BE THE STATE
C OR REGION CODE, AND THE SECOND TWO DIGITS, THE SECTION WITHIN THE
C STATE OR REGION.
C
C INTEGER# STATE, SECT
C INTEGER# TENTS,FIFTY
C DIMENSION INPUT(25,8), MPUT(4), LPUT(50,4)
C DIMENSION INOUT(200), MYOUT(200)
C DEFINE FILE 301(40,800), LK30), 31(11,800,LK31)
C DATA IBLNK/ 1/
C K30= 1
C K31= 1
C FIFTY = 0
d0 10 I = 1,25
D0 11 J = 1,4
11 INPUT(I,J) = IBLNK
D0 10 J = 5,8
D0 10 I = 1,4
10 INPUT(I,J) = 0
C FORMAT ONE (1) IS USED FOR COMPETENCIES, ORGANIZATIONS, AND FUNCTIONS
C FORMAT TWO (2) IS USED FOR LOCATION AND REGION PREFERENCES.
C 1 FORMAT(14,4A4)
C 2 FORMAT (12,4A4)
C 5 FORMAT (* 121*,15)
C 6 FORMAT(* 154*,15)
C TENTS = 0
C BEGIN BLOCKING AND OUTPUTING OF COMPETENCIES, ORGANIZATIONS AND
C FUNCTIONS IN MULTIPLES OF 25 BY CODE NUMBERS.
D0 20 I = 1,40
19 READ(1,1) NUMB, MPUT
NUMB = NUMB - TENTS
IF (NUMB < 67-25) GO TO 24
D0 21 J = 1,4
21 INPUT(NUMB,J) = MPUT(J).
GO TO 19
24 TENTS = TENTS + 25
C SET UP THE OUTPUT ARRAY AND WRITE ON DISK.
C ME = 0
D0 121 LE = 1,25
D0 121 KE = 1,8
ME = ME + 1
121 INPUT(ME) = INPUT(LE,KE)
WRITE(*5,*) ME
WRITE(*30,K30) INOUT
D0 27 IX = 1,25
D0 27 JX = 1,4
27 INPUT(IX,JX) = IBLNK
IF (IX.EQ.40) GO TO 20
NUMH = NUMH - 25
DO 29 JX = 1,4
29 INPUT(NUMH,JX) = MPUT(J)
C CONTINUE
C END GROUP ONE.
C BEGIN BLOCKING AND OUTPUTING OF LOCATION PREFERENCE CODES IN
C IN MULTIPLES OF 50 BY CODE NUMBER.
C BLANK AND ZERO LPUT ARRAY
DO 40 I = 1,50
LPUT(I,1) = IBNKL
DO 40 J = 2,4
40 LPUT(I,J) = 0
DO 50 I = 1,11
51 READ(1,2) STATE,SECT,IABRV.
C USE STATE AND SECT TO DETERMINE ARRAY POSITIONS.
NUMB = ((STATE - 1) * 10 + SECT) - FIFTY
IF (NUMB .GT. 50) GO TO 54
LPUT(NUMB,1) = IABRV
GO TO 51
54 FIFTY = FIFTY + 5C
ME = 0
C SET UP THE OUTPUT ARRAY AND WRITE ON DISK.
DO 154 LE = 1,50
DO 154 KE = 1,4
ME = ME + 1
154 MYOUT(ME) = LPUT(LE,KE)
WRITE(3,M6) ME
WRITE(31,K31) MYOUT
DO 57 IX = 1,50
57 LPUT(IX,1) = IBNKL
IF (IX.EQ. 11) GO TO 50
NUMB = NUMB - 50
LPUT(NUMB,1) = IABRV
50 CONTINUE
C END GROUP TWO
STOP 15
END
//GO.SYSPRINT DD SYSOUT=A
//GO.FT31FO01 DD DSN=COP012PC.L0CRGDIR.REL.UNIT=SYSDA,VDL=SER=222222, X
// DISP=OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//GO.FT31FO01 DD DSN=COP012PC.COMPORFN.REL.UNIT=SYSDA,VDL=SER=222222, X
// DISP=OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//GO.SYSIN DD *
AN INFORMATION SYSTEM FOR EDUCATIONAL PLACEMENT

by

RICHARD ELLIS LONSINGER

B.S., Kansas State University, 1965

—

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971
AN INFORMATION SYSTEM FOR EDUCATIONAL PLACEMENT

The design of an information storage and retrieval system to be used by the Educational placement Section, Kansas State University Career Planning and Placement Center, is presented. The design presented was determined to best solve the problem of handling registration information on job candidates seeking employment in the field of Education. The system utilizes student data maintained by the Office of Admissions and Records for creating most of the Educational Placement Master Data File.

Flexibility in the information system was achieved by providing for master record retrieval by either registrant social security numbers or by utilizing the codes for teaching competencies, organizations and functions, and job location preferences. As master data records are updated, the data fields corresponding to the previously mentioned coded variables are linked from one master record to another by using a backward and forward chaining field in each record. This double-linking technique provides for rapid retrieval of data by link variable arguments and permits new records to be added to or deleted from the chain very easily.

Finally, the details of the data files and the programs for initializing and updating them are discussed.