

**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 6791

RICHARD ELLIS LONSINGER

B. S., Kansas State University, 1965

—

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

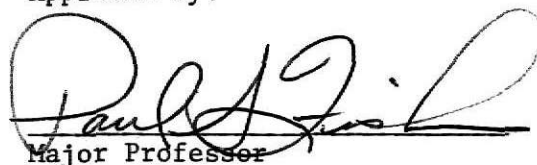
MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Approved by:


Major Professor

LD
2668
R4
1971
L65
C.2

TABLE OF CONTENTS

	PAGE
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

	PAGE
APPENDIX E (Alumnus File Activation Input Card Format for FILEDEVL Program)	25
APPENDIX F (FILEDEVL Program Listing)	27
APPENDIX G (CREATE Program Listing)	41
APPENDIX H (DIRCREA Program Listing)	47

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 LOCRGDIR, 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 variable information. The link variable is used to store the record number of the last record in the master (TOTALREC) file that used this location preference. The array is reblocked and written to the LOCRGDIR file when processing 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).

[illegible]

Figure 2. Input Card Format for Coding Competencies, Organizations, and Functions.

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 accomplished using any higher level programming language having an unformatted input/output feature.

BIBLIOGRAPHY

Germain, Clarence B. Programming the IBM 360. Englewood Cliffs, New Jersey: Prentice Hall, 1967.

IBM System/360 Operating System Supervisor and Data Management Services.
IBM Form No. C28-6646.

Knuth, Donald E. The Art of Computer Programming. Vol. 1. Reading, Massachusetts: Addison-Wesley, 1968.

Lefkovitz, David. File Structures for On-line Systems. New York: Spartan Books, 1969.

Appendix A.

TOTALREC Record Variable List

<u>VARIABLE</u>	<u>MODE</u>	<u>DESCRIPTION</u>
RWSSN	I4	Social Security Number
RWNAME(5)	5A4	Registrant's Name
DOB(3)	3I2	Registrant's date of birth
NUMCUR	I2	Numeric code for Registrant's curriculum (A & R code)
RWALCU	A4	Alpha code for Registrant's curriculum (A & R code)
SCHOOL	I2	Numeric code for Registrant's school (A & R code)
CLASS	I2	Numeric code for Registrant's class (A & R code)
RESCOD	I2	Numeric code for Registrant's residence status (A & R code)
PLTEA	I2	Numeric code indicating Registrant's plans to teach 1 - yes, 2 - no, 3 - undecided
PLGRAD	I2	Numeric code indicating Registrant's plans to attend graduate school, 1 - yes, 2 - no, 3 - undecided
REGPC	I2	Numeric code indicating that Registrant has registered at the Placement Center, 1 - yes, 2 - no
RWZIP	I4	Registrant's permanent address, zip code
DATGRA(2)	2I2	Registrant's anticipated graduation date
RWCIT	A4	Registrant's citizenship
VET	I2	Registrant's veteran status (A & R code)
SEXMAR	I2	Registrant's sex and marital code (A & R code)
RWLPHO(2)	2A4	Registrant's local phone number
RWLADR(5)	5A4	Registrant's local address (street or box number)
RWLCS(5)	5A4	Registrant's local address (city/state)
RWLZIP	I4	Registrant's local zip code
RWPAD(5)	5A4	Registrant's permanent address (street or box number)

RWPCS(5)	5A4	Registrant's permanent address (city/state)
RWPHON(2)	2A4	Registrant's permanent phone number
TYPESRG	I2	Registrant's type of Placement Center Registration (0 - none, 1 - Educational Placement, 2 - B & I, 3 - Both)
FILCOD	I2	Type of file code (1 - actively seeking employment, 2 - alumnus seeking employment, 3 - inactive file, 4 - registered-not actively seeking employment)
TEALV1(3)	3I2	First Teaching Level TEALV1(1) - Level Code TEALV1(2) - Backward link to previous record with same level 1 code TEALV1(3) - Forward link to next record with same level 1 code
TEALV2(3)	3I2	Second Teaching Level TEALV2(1) - Level code TEALV2(2) - Backward link (not presently used) TEALV2(3) - Forward link (not presently used)
MAJARA(3)	3I1	First Competency MAJARA(1) - Competency code MAJARA(2) - Backward link to previous record with same first competency code MAJARA(3) - Forward link to next record with same first competency code
SUPFD1(3)	3I2	Supporting competency one SUPFD1(1) - Competency code SUPFD1(2) - Backward link (not presently used) SUPFD1(3) - Forward link (not presently used)
SUPFD2(3)	3I2	Supporting competency two SUPFD2(1) - Competency code SUPFD2(2) - Backward link (not presently used) SUPFD2(3) - Forward link (not presently used)
SUPFD3(3)	3I2	Support competency three SUPFD3(1) - Competency code SUPFD3(2) - Backward link (not presently used) SUPFD3(3) - Forward link (not presently used)
LOCPR1(3)	3I2	Location preference one LOCPR1(1) - Location preference code LOCPR1(2) - Backward link to previous record with the same location preference code LOCPR1(3) - Forward link to next record with the same location preference code

LOCPR2(3)	3I2	Location preference two LOCPR2(1) - Location preference code LOCPR2(2) - Backward link (not presently used) LOCPR2(3) - Forward link (not presently used)
LOCPR3(3)	3I2	Location preference three LOCPR3(1) - Location preference code LOCPR3(2) - Backward link (not presently used) LOCPR3(3) - Forward link (not presently used)
LOPRCD	I2	Location preference code (0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)
MUSTCS	4A4	City/state for "1" location preference
AVAIL	I2	Date available code (1 - December, 2 - Jan-Feb, 3 - Jun-July, 4 - Aug-Sept)
HWCOMB	I2	Husband-Wife teaching combination code (1 - yes, 2 - no)
MINSAL	I2	Minimum salary acceptable - \$100's
DGYR	I2	Degree and year awarded code (highest award) (1__ Bachelor's, 2__ Master's, 3__ Master's + 30, 4__ Doctorate's Blank space will contain the last two digit of the year the degree was awarded)
YREXPR(3)	3I2	Years of employment experience YREXPR(1) - Years of teaching experience YREXPR(2) - Years of education administration experience YREXPR(3) - Years of non-education experience
POGRPL	I2	Post Graduation Plans (Code 0 - Code 7 --Diploma Receipt, Code 8 - seeking non-education employment)
INEMPL(4)	4A4	Employer (from diploma receipt, inactivate card, or registered-inactive card)
INPOSI(3)	3A4	Position (from diploma receipt, inactivate card, or registered-inactive card)
INADDR(5)	5A4	Address of employer (from diploma receipt, inactive card, or registered-inactive card)
INSALY	I4	Starting Salary (from diploma receipt, inactivate card, or registered-inactive card)

INMOSR	I2	Months of service for which the starting salary is received
EDEMPL(3)	3I2	Educational Employment Information EDEMPL(1) - Level code EDEMPL(2) - Years of education experience code EDEMPL(3) - Return to former position code

Appendix B.

Inactivation Input Card Format for FILEDEVL Program

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

Appendix C.

Registered - Inactive Input Card format for
FILEDEVL Program

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

Appendix D.

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

Card 1.

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

Card 2.

<u>Card Column</u>	<u>FORTTRAN Mode</u>	<u>Program Variable</u>	<u>Description</u>
1 - 4	A4	INCHAR	Card Identifier - ACTV
5 - 9	I4	INZIP	Local ZIP Code
10 - 17	2A4	INLPHN	Local Phone Number
18	I2	INMAR	Marital Status
19 - 21	I2	INDGYR	Degree & Year Awarded (See DGYR description, App. A)
22 - 27	3I2	INYREX	Employment Experience (See YREXPR description, App. A)
28	I2	INHSWF	Husband-Wife Teaching Combination Code (1 - yes, 2 - no)
29 - 31	I2	INMINS	Minimum Salary Acceptable - \$100's
32	I2	INAVIL	Date Available Code (1 - Dec, 2 - Jan-Feb, 3 - Jun-Jul, 4 - Aug-Sep)

Appendix D (Continued).

<u>Card Column</u>	<u>FORTTRAN Mode</u>	<u>Program Variable</u>	<u>Description</u>
33 - 44	4I2	INCOMP	Teaching Competencies
45 - 50	2I2	INSTYP	Institution Type
51 - 62	3I2	INLOCP	Location Preferences Codes
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.

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

Card 2.

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

Appendix E (Continued).

<u>Card Column</u>	<u>FORTTRAN Mode</u>	<u>Program Variable</u>	<u>Description</u>
75 - 79	-	-	Spaces
80	I4	ICDTP	Card Sequence Number - "2"
Card 3.			
<u>Card Column</u>	<u>FORTTRAN Mode</u>	<u>Program Variable</u>	<u>Description</u>
1 - 4	A4	INCHAR	Card Identifier - ALUM
5 - 10	3I2	INYREX	Employment Experience (See YREXPR description, App. A)
11	I2	INHSWF	Husband-Wife Teaching Combination Code (1 - yes, 2 - no)
12 - 14	I2	INMINS	Minimum Salary Acceptable - \$100's
15	I2	INAVIL	Date Available Code (See AVAIL description, App. A)
16 - 27	4I2	INCOMP	Teaching Competencies
28 - 33	2I2	INSTYP	Institution Type
34 - 45	3I2	INLDCP	Location Preferences Codes
46	I2	INLPRE	Location Preference (0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)
47 - 62	4A4	MUSTCS	City/State for "a necessity" Location preference
63 - 79	--	--	Spaces
80	I4	ICDTP	Card Sequence Number - "3"

Appendix F.

Listing of FILEDEVL Program.

```

// EXEC FORTGCLG,PARM.FCRT='MAP'
//FORT.SYSIN DD *
C FILEDEVL PROGRAM FOR EDUCATIONAL PLACEMENT DATA FILES,
C BY R. E. LUNSFORD, AUGUST 1970. THIS PROGRAM ACCEPTS
C THE FOLLOWING TYPES OF DATA CARDS IN ANY ORDER:
C 1. ACTIVATION CARDS -- ACTIV IN CC 1-4
C 2. ALUMNI CARDS -- ALUM IN CC 1-4
C 3. INACTIVATION CARDS -- INAC IN CC 1-4
C 4. REGISTERED - INACTIVE CARDS -- RINA IN CC 1-4
C SEE USER MANUAL FOR COMPLETE DEFINITION OF THE ABOVE
C DATA CARDS
C
C MAIN ARRAYS USED BY THIS PROGRAM
C INDEX(3400,2) -- THIS ARRAY CONTAINS THE SSN OF THE
C INDIVIDUALS IN THE TOTALREC FILE AND A POINTER TO
C THE LOCATION OF ALL THE INFORMATION PRESENTLY STORED
C FOR THE INDIVIDUAL IN THE TOTALREC FILE
C ICOMP(1000,8) -- THIS ARRAY CONTAINS THE BACKWARD LINK
C INFORMATION FOR COMPETENCIES, TYPE OF ORGANIZATION AND
C NON-TEACHING TITLES. BACKWARD LINKS ARE PROVIDED FOR UP TO
C FOUR LEVELS OF COMPETENCIES AND UP TO TWO TEACHING AREAS
C LOCAT(550,4) -- THIS ARRAY CONTAINS THE BACKWARD LINK
C INFORMATION FOR LOCATION PREFERENCES. LINKS ARE PROVIDED
C FOR UP TO THREE LOCATION PREFERENCES.
C
C SEE APPENDIX F FOR A COMPLETE LISTING OF THE VARIABLES USED BY
C THIS PROGRAM.
C
C THIS PROGRAM USES THE REREAD SUBROUTINE TO REFORMAT DATA AFTER IT
C HAS BEEN INPUTTED UNDER CONTROL OF A DIFFERENT FORMAT.
C SEE THE KSU COMPUTING CENTER NOTICE NUMBER 151 FOR ADDITIONAL
C INFORMATION ABOUT THIS SUBROUTINE.
C
C INTEGER*4 RWSSN, RWNAME(5),RWALCU, RWZIP, RWKIT, RWLPHO(2),
C 1RWLADR(5), RWLCS(5), RWLZIP, RWPAD(5), RWPCS(5), RWPHUN(2),
C 2MUSTCS(4), BUFL(41), TLBKLK, TLFLRK, MAJBLK, MAJFRL, LOCBLK,
C 3LOCFLK
C
C INTEGER*2 DOB(3), NUMCUR, SCHOOL, CLASS, RESCOD, PLTFA, PLGRAD,
C 1 REGPC, DATGRA(2), VET, SEXMAR, TEALV1(3), TEALV2(3), MAJARA(3),
C 2 SUPFD1(3), SUPFD2(3), SUPFD3(3), LOCPR1(3), LOCPR2(3), LOCPR3(3),
C 3FILCOD, AVAIL, HWCOMB, MINSAL, DGYR, YREXP(3), TYPCRG, LOPRCU, PDGRPL,
C 4BUF2(18), INDOB(3), INDOGR, INYREX(3), INHSWF, INMINS, INAVIL,
C 5INSEX, EDEMP(3), INMOSR
C
C DIMENSION INAME(5), INADDR(5), INEMPL(4), INCTYS(5),
C 1INLPHN(2), INCOMP(4), INSTYP(2), INLOCP(3), INPUSI(3)
C
C DIMENSION INPUT(80), ICOMP(1000,8), INOUT(200), LOCAT(550,4)
C
C COMMON INDEX(3400,2), L1, L3
C
C EQUIVALENCE (INPUT(1), INOUT(1))
C
C DATA IACTV/'ACTV', IRINA/'RINA', IALUM/'ALUM', INAC/'INAC',
C DATA IBLNK/' '
C
C DEFINE FILE 30(40,800,L,K30), 31(11,800,L,K31)
C DEFINE FILE 20(3400,320,L,K20), 21(8,3520,L,K21)
C
C SET UP INDEX ARRAY FROM THE DATA SET
C
C ZERO INDEX
C DO 10 J = 1,2
C DO 10 I = 1,3400
C 10 INDEX(I,J) = 0
C
C SET ALL THE ASSOCIATED VARIABLES TO ONE

```

```

K20 = 1
K31 = 1
K30 = 1
K21 = 1
L1 = 1
C READ A BLOCK OF RECRD FROM THE INDEXREC FILE.
C DEBLOCK INTO THE INDEX ARRAY
14 READ (21,K21) INPUT
DO 15 I = 1,880,2
J = I + 1
IF (INPUT(I)) 20,20,16
20 INDEX(L1,I) = INPUT(I)
GO TO 22
16 INDEX(L1,I) = 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
IFIXIT = 0
J25 = 25
I25 = 1
C READ A BLOCK OF COMPETENCIES, ORGANIZATIONS, AND FUNCTIONS FROM
C THE COMPUNEN FILE. DEBLOCK INTO THE ICOMP ARRAY.
DO 32 I = 1,40
READ(30,K30) INOUT
ME = 0
DO 33 J = I25,J25
DO 33 K = 1,8
ME = ME + 1
33 ICOMP(J,K) = INOUT(ME)
I25 = I25 + 25
J25 = J25 + 25
C END THIS SET UP
C BEGIN SET UP OF LOCATION PREFS. AND REGION PREFS.
I50 = 1
J50 = 50
DO 42 I = 1,11
C READ A BLOCK OF LOCATIONS AND REGIONS FROM THE LOCRGDIR FILE.
C DEBLOCK INTO THE LOCAT ARRAY.
READ(31,K31) INOUT
ME = 0
DO 43 J = I50,J50
DO 43 K = 1,4
ME = ME + 1
43 LOCAT(J,K) = INOUT(ME)
I50 = I50 + 50
J50 = J50 + 50
C END THIS SET UP
C USE THE REREAD SUBROUTINE TO DETERMINE WHICH FORMAT TO USE
CALL REREAD
5 FORMAT (A4)
100 READ(1,5,END=1950) ICHAR
IF (ICCHAR .EQ. IACTIV) GO TO 600
IF (ICCHAR .EQ. IIRINA) GO TO 400
IF (ICCHAR .EQ. IALUM) GO TO 800
IF (ICCHAR .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(99,201) INCHAR,INSSN,INCODE,INSALY,INMUSR,INEMPL,INPOSI,
  1INADDR
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
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
  GO TO 1210
1212 LMAX = MAX
  GO TO 1214
1210 CONTINUE
  WRITE(3,1213) INSSN
1213 FORMAT(' SSN ',I10,' NOT IN INDEXREC FILE')
  GO TO 100
1214 INDEX(LMAX,2) = LREC
  IRECNO = LREC
  IF (IFIXIT) 207,207,418
207 K20 = IRECNO
  READ (20,K20) BUFF1,TYPECRG,FILCOD,TEALV1,TEALV2,MAJARA,SUPFD1,
    1 SUPFD2,SUPFD3,LOCPR1,LOCPR2,LOCPR3,MUSTCS,AVAIL,HWCOMB,MINSAL,
    2 DGYR,YKEXPR,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. ',I10,3A4)
  WRITE(3,261) INSSN,BUFF1(2),BUFF1(3),BUFF1(4)
  GO TO 100
260 CONTINUE
C SAVE PART OF THE NAME
  INAME(1) = BUFF1(2)
  INAME(2) = BUFF1(3)
  INAME(3) = BUFF1(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. IRECN0)
1  ICOMP(TEALV1(1),5) = TEALV1(2)
IF (ICOMP(MAJARA(1),5) .EQ. IRECN0)
1  ICOMP(MAJARA(1),5) = MAJARA(2)
ITEMP1 = LOCPR1(1)
IF (ITEMP1) 301,301,302
302 NTEMP1 = ITEMP1/100
ITEMP1 = (ITEMP1-1)*10 + (ITEMP1-(NTEMP1*100))
IF (LOCAT(ITEMP1,2) .EQ. IRECN0)
1 LOCAT(ITEMP1,2) = LOCPR1(2)
301 CONTINUE
C SAVE THE BACKWARD AND FORWARD LINK VARIABLES
C TEACHING LEVEL ONE BACKWARD LINK
TLBKLV=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 PREF ONE BACKWARD LINK
LOCBLK = LOCPR1(2)
C LOCATION PREF ONE FORWARD LINK
LOCFLK = LOCPR1(3)
C SET POST GRADUATION PLANS CODE
POGRPL = INCODE
C SET FILE TYPE CODE TO INACTIVE
FILCOD = 3
C SET DA POINTER TO RECORD AND WRITE UPDATED RECORD
K20 = IRECN0
WRITE(20,K20) BUFF1,TPCRG,FILCOD,TEALV1,TEALV2,MAJARA,SUPFD1,
1 SUPFD2,SUPEU3,LOCPR1,LOCPR2,LOCPR3,MUSTCS,AVAIL,HWCMB,MINSAL,
2 DGYR,YREXP,POGRPL,INEMPL,INPOSI,INADDR,INSALY,INMOSR
C IF THIS BACKWARD LINK IS ZERO, BRANCH TO THE NEXT
IF (TLBKLV) 230,230,231
231 K20 = TLBKLV
READ(20,K20) BUFF1,TPCRG,FILCOD,TEALV1
C SET THE FORWARD LINK OF THE BACKWARD LINK EQ TO THE FORWARD
C LINK OF THIS RECORD
TEALV1(3) = TLFRLK
K20 = TLBKLV
WRITE(20,K20) BUFF1,TPCRG,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,TPCRG,TEALV1
C SET THE BACKWARD LINK OF THIS RECORDS BACKWARD LINK EQUAL TO THE
C BACKWARD LINK OF THE INACTIVATED RECORD
TEALV1(2) = TLBKLV
K20 = TLFRLK
WRITE(20,K20) BUFF1,TPCRG,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, TPCRG, FILCOD, TEALV1,TEALV2,MAJARA
MAJARA(3) = MAJFRL
K20 = MAJBLK
WRITE(20,K20) BUFF1,TPCRG,FILCOD,TEALV1,TEALV2,MAJARA
C IF MAJOR AREA FORWARD LINK IS ZERO, BRANCH TO THENEXT
234 IF (MAJFRL) 236,236,237

```

```

237 K20 = MAJFRL
   READ(20,K20)  BUFF1,TPPCRG,FILCOD,TEALV1,TEALV2,MAJARA
   MAJARA(2) = MAJULK
   K20 = MAJFRL
   WRITE(20,K20)  BUFF1,TPPCRG,FILCOD,TEALV1,TEALV2,MAJARA
C  IF LOCATION PREF BACKWARD LINK IS ZERO, BRANCH TO NEXT
236 IF (LOCBLK) 238,238,239
239 K20 = LOCBLK
   READ(20,K20)  BUFF1,TPPCRG,FILCOD,BUF2,LOCPR1
   LOCPR1(3) = LOCFLK
   K20 = LOCBLK
   WRITE(20,K20)  BUFF1,TPPCRG,FILCOD,BUF2,LOCPR1
C  IF LOCATION PREF FORWARD LINK IS ZERO, BRANCH TO THE NEXT
238 IF (LOCFLK) 240,240,241
241 K20 = LOCFLK
   READ(20,K20)  BUFF1,TPPCRG,FILCOD,BUF2,LOCPR1
   LOCPR1(2) = LOCBLK
   K20 = LOCFLK
   WRITE(20,K20)  BUFF1,TPPCRG,FILCOD,BUF2,LOCPR1
240 WRITE(3,242) INSSN,INAME(1),INAME(2),INAME(3)
242 FORMAT(' INACTIVATION COMPLETE FOR NUMBER ',I10,1X,3A4)
   GO TO 100
C
C
201 FORMAT(A4,I9,I1,I5,I2,4A4,3A4,5A4)
202 FORMAT(' INPUT ERROR IN CARD ',A4,I9,' -- NOT IN INDEX FILE')
220 WRITE(3,202) INCHAR,INSSN
   GO TO 100
C
C
C  END RECORD INACTIVATION
C
C  BEGIN REGISTERED - INACTIVE FILE PROCESSING
C
400 READ(99,201) INCHAR,INSSN,INCODE,INSALY,INMOSR,INEMPL,
   1 INPOSI,INADDR
C  GO PICK UP THE RECORD NUMBER
   CALL BINSCH(INSSN,IRECNO)
C  IF RECORD NOT FOUND OUTPUT ERROR MESSAGE
   IF (IRECNO) 420,420,405
405 K20 = IRECNO
   READ (20,K20)  BUFF1,TPPCRG,FILCOD,TEALV1,TEALV2,MAJARA,SUPFD1,
   1 SUPFD2,SUPFD3,LOCPR1,LOCPR2,LOCPR3,MUSTICS,AVAIL,HWCMB,MINSAL,
   2 DGYR,YREXPR,POGRPL
   FILCOD = 4
   K20 = IRECNO
   WRITE(20,K20)  BUFF1,TPPCRG,FILCOD,TEALV1,TEALV2,MAJARA,SUPFD1,
   1 SUPFD2,SUPFD3,LOCPR1,LOCPR2,LOCPR3,MUSTICS,AVAIL,HWCMB,MINSAL,
   2 DGYR,YREXPR,POGRPL,INEMPL,INPOSI,INADDR,INSALY,INMOSR
   WRITE(3,406) INSSN,BUFF1(2),BUFF1(3),BUFF1(4)
406 FORMAT(' REGISTERED - INACTIVE FILE ESTABLISHED FOR NUMBER ',I10,
   11X,3A4)
   GO TO 100
420 WRITE(3,202) INCHAR, INSSN
   GO TO 100
C
C  END REGISTERED - INACTIVE FILE PROCESSING
C
C  BEGIN FILE ACTIVATION PROCESSING
C
C  ACTIVATE THE FILE FOR THE INDIVIDUAL.

```

```

403 FORMAT(A4,I5,2A4,I1,I3,3I2,I1,I3,I1,I3,3I4,I1,4A4)
401 FORMAT (6A4,I9,I1,3I2,10A4)
402 FORMAT(A4,75X,I1)
600 READ (99,401) INCHAR, INAME, INSSN, INSEX, INDOB, INADDR, INCTYS
C SET THE ALUMNUS FILE PROCESSING SWITCH OFF
      IALSWH = 0
C CHECK CARD CODE AND IF SECOND CARD
      READ(1,402) ICHAR, ICDTP
      IF (ICCHAR .NE. IACTV) GO TO 410
      IF (ICDIP .NE. 2) GO TO 410
      READ (99,403) INCHAR, INZIP, INLPHN, INMAR, INDOYR, INYREX, INHNSW,
      LINMNS, INAVIL, INCMP, INSTYP, INLOCP, INLPRE, MUSICS
C FIND THE TOTAL RECORD NUMBER FOR THIS SSN
      CALL BINSCH(INSSN,IRECNO)
      IF (IRECNO) 420,420,419
419 IF (IRECNO .GT. LEND) GO TO 500
      GO TO 418
500 IFIXIT = 1
      GO TO 1200
418 K20 = IRECNO
      READ(20,K20) RWSSN, RNAME, DOB, NUMCUR, RWALCU, SCHOOL, CLASS,
      IRESCOD, PLTEA, PLGRAD, REGPC, RWPZIP, DATGRA, RWCIT, VET, SEXMAR,
      2RWLPHN, RWLADR, RWLCS, RWLZIP, RWPAD, RWPCS, PWPHON, TYPCRG, FILCOD,
      3TEALV1, TEALV2, MAJARA, SUPFD1, SUPFD2, SUPFD3, LOCPR1, LOCPR2,
      4LOCPR3, LOCPR4
455 CONTINUE
C SET THE BACKLINK IN THIS RECORD FOR TEACHLEVEL ONE, MAJOR AREA
      AND LOCATION PREFERENCE ONE
      ITEMPL = INCOMP(1)
      MAJARA(2) = ICOMP(ITEMPL,5)
      ICOMP(ITEMPL,5) = IRECNO
C
      ITEMPL = INSTYP(1)
      TEALV1(2) = ICOMP(ITEMPL,5)
      ICOMP(ITEMPL,5) = IRECNO
C
C
C DECODE THE LOCATION PREFERENCE FOR LOCAT ARRAY
      ITEMPL = INLOCP(1)
      NITEMPL = ITEMPL/100
      ITEMPL = ((NITEMPL-1)*10)+(ITEMPL-(NITEMPL*100))
      LOCPR1(2) = LOCAT(ITEMPL,2)
      LOCAT(ITEMPL,2) = IRECNO
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
C MAJOR AREA
      ITEMPL = MAJARA(2)
C TEACHING LEVEL
      NITEMPL = TEALV1(2)
C LOCATION PREF ONE
      MITEMPL = LOCPR1(2)
C
C SET UP THE UPDATED INFO FOR WRITE
C TEACHING LEVELS ONE AND TWO CODES
      TEALV1(1) = INSTYP(1)
      TEALV2(1) = INSTYP(2)
C MAJOR AREA CODE

```

```

C  MAJARA(1) = INCOMP(1)
C  SUPPORTING FIELDS ONE, TWO AND THREE CODES
  SUPFD1(1) = INCOMP(2)
  SUPFD2(1) = INCOMP(3)
  SUPFD3(1) = INCOMP(4)
C  LOCATION PREF ONE, TWO, AND THREE CODES
  LOCPR1(1) = INLOCP(1)
  LOCPR2(1) = INLOCP(2)
  LOCPR3(1) = INLOCP(3)
C  CHECK FOR UPDATED DOB
  IF (INDOB(1) .EQ. 0 .OR. INDOB(2) .EQ. 0 .OR. INDOB(3) .EQ. 0)
    1 GO TO 424
C  UPDATE THE DATE OF BIRTH
  DO 425 JX1=1,3
  425 DOB(JX1) = INDOB(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 PREF.
  LOPRCD = INLPRE
C  CHECK FOR PROCESSING ALUMNUS 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
  TYPCRG = 1
  FILCOD = 1
  475 K20 = IRECN0
  WRITE(20,K20) RWSSN, INAME, DOB, NUMCUR, RWALCU, SCHOOL, CLASS, RESCUD,
    1PLTEA, PLGRAD, REGPC, RWPZIP, DATGRA, RWCIT, VET, SEXMAR, INLPHN, INADDR,
    2INCTYS, INZIP, RWPAC, RWPCH, RWPCHON, TYPCRG, FILCOD, TEALV1, TEALV2,
    3MAJARA, SUPFD1, SUPFD2, SUPFD3, LOCPR1, LOCPR2, LOCPR3, LOPRCD, MUSTCS,
    4INAVIL, INHSWF, INMINS, INDOYR, INYREX
C  CHANGE THE FRONT LINKS IN THE BACK LINKS
C  TEACHING LEVEL
  IF (NTEMP1) 430,430,431
  431 K20 = NTEMP1
  READ(20,K20) BUFF1, TYPCRG, FILCOD, TEALV1
  TEALV1(3) = IRECN0
  K20 = NTEMP1
  WRITE(20,K20) BUFF1, TYPCRG, FILCOD, TEALV1
C  MAJOR AREA
  430 IF (ITEMP1) 432,432,433
  433 K20 = ITEMP1
  READ(20,K20) BUFF1, TYPCRG, FILCOD, TEALV1, TEALV2, MAJARA
  MAJARA(3) = IRECN0
  K20 = ITEMP1
  WRITE(20,K20) BUFF1, TYPCRG, FILCOD, TEALV1, TEALV2, MAJARA
C  LOCATION PREFERENCE
  432 IF (MTEMP1) 440,440,435
  435 K20 = MTEMP1
  READ(20,K20) BUFF1, TYPCRG, FILCOD, BUF2, LOCPR1

```

```

LOCPR1(3) = IRECNC
K20 = MTEMP1
WRITE(20,K20) BUFF1,TPCRG,FILCOD,BUF2,LOCPR1
C CHECK FOR PROCESSING ALUMNUS CARDS
440 IF (IALSWH) 445,445,870
445 WRITE(3,446) RWSSN,RWNAME(1),RWNAME(2),RWNAME(3)
446 FORMAT(' FILE ACTIVATED FOR NUMBER ',I10,I1,3A4)
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,I9,I1,3I2,10A4)
805 FORMAT(A4,3I2,I1,I3,I1,6I3,3I4,I1,4A4)
803 FORMAT(A4,I5,I2A4,I5,2A4,I1,I3)
C SET THE ALUMNUS FILE PROCESSING SWITCH TO ON (1)
800 IALSWH = 1
READ(99,801) INCHAR,INAME,INSSN,INSEX,INDOB,INADDR,INCTYS
C CHECK CARD CODE AND IF SECOND CARD
READ(1,402) ICHAR,ICDTP
IF(ICHAR.NE.IALUM) GO TO 810
IF(ICDTP.NE.2) GO TO 810
READ(99,803) INCHAR,INZIP,INLPHN,RWPAD,RWPCS,RWPZIP,RWPHON,INMAR,
IINDGYR
C CHECK CARD CODE AND IF THIRD CARD
READ(1,402) ICHAR,ICDTP
IF(ICHAR.NE.IALUM) GO TO 811
IF(ICDTP.NE.3) GO TO 811
READ(99,805) INCHAR,INYREX,INHSWF,INMINS,INAVIL,INCOMP,INSTYP,
IINLOCP,IINLPRE,MUSTCS
C GO FIND THE TOTAL 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
SSN IS IN ERROR.')
GO TO 100
810 WRITE(3,411) INAME,INSSN
READ(1,402) ICHAR,ICDTP
GO TO 100
811 WRITE(3,411) INAME,INSSN
GO TO 100
C GO ADD 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
TPCRG=1
FILCOD=2
C GO SET UP LINKS AS SPECIFIED
GO TO 455
865 CONTINUE
C SET UP THOSE VARIABLES NORMALLY TAKEN FROM THE A&R FILE
NUMCUR=0
RWALCU=IBLNK
SCHOOL=0
CLASS=0

```

```

RESC00=0
PLTEA=0
PLGRAD=0
REGPC=0
DATGRA(1)=0
DATGRA(2)=0
RWCIT=IBLNK
VET=0
SEXMAR=INSEK
C GO WRITE THE RECORD AND SET THE LINKS
GO TO 475
C
871 FORMAT(' ALUMNUS FILE ACTIVATED FOR NUMBER ',I10,Ix,3A4)
870 WRITE(3,871) INSSN,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(LI,1)) 2010,2010,2005
2010 INPUT(I) = INDEX(LI,1)
GO TO 2015
2005 INPUT(I) = INDEX(LI,1)
INPUT(J) = INDEX(LI,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.
I25 = 1
J25 = 25
DO 2032 I=1,40
ME = 0
DO 2021 LE=I25,J25
DO 2021 KE=1,8
ME=ME+1
2021 INOUT(ME) = ICOMP(LE,KE)
WRITE(30,K30) INOUT
I25 = I25+25
2032 J25 = J25+25
C END ICOMP RESTORE
C RESTORE THE LOCAT ARRAY BY BLOCKING AND WRITING ON THE DISK.
J50 = 50
I50 = 1
DO 2042 I=1,11
ME = 0
DO 2043 J=I50,J50
DO 2043 K=1,4
ME = ME+1
2043 INOUT(ME) = LOCAT(J,K)
WRITE(31,K31) INOUT
I50 = I50+50

```

```

2042 J50 = J50+50
      WRITE(3,2052)
2052 FORMAT(' NORMAL END OF JOB')
      STOP 2
      END
      SUBROUTINE BINSCH (ISSN,IRECNO)
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 - I
      1 CONTINUE
      IF (ISSN - 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
      4 IF (INCR .EQ. 1) GO TO 16
      IF (INCR .LT. 1) GO TO 5
      I = I + INCR
      INCR = (INCR + 1) / 2
      IF (I .GT. LEND) I = I - INCR
      GO TO 1
      3 IRECNO = 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
      IRECNO = 0
      L3 = I
      RETURN
      7 IRECNO = 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(IL3,1) .LT. ISSN) GO TO 20
IF (INDEX(IL3,1) .GT. ISSN) GO TO 50
IRTN = -1
RETURN
20 IL3 = L3 + 2
24 INDEX(IL1,1) = INDEX(IL1-1,1)
INDEX(IL1,2) = INDEX (IL1-1,2)
IF (IL1 .LE. IL3) 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,SYSPRINT UD SYSOUT=A
//FT20F001 DD DSN=CDP12PC.TOTALREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=320) X
//FT21F001 DD DSN=CDP12PC.INDEXREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=3520) X
//FT30F001 DD DSN=CDP12PC.COMPOFN.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800) X
//FT31F001 DD DSN=CDP12PC.LOCRGDIR.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//GO,SYSDA DD *
INAC515348145 100001UAL SYS ANAL DENVER, COLO
ACTVLONINGER,RICHARD 5153481451122339BOX 114 LOUISVILLE,KANSAS
ACTV664504567298 2270.0 0 0 760 1601 2
RINA51346398771000012KSU COMP CNTR SYS ANAL MANHATTAN,KANSAS
ALUMORINETT,SHERRY D. 49552535720420489046 WALNUT ST. KANSAS CITY, MO
ALUM DE31832 9046 WALNUT KANSAS CITY, MO DE318321170
ALUM 0 0 02 04760 010 16022508 IKANSAS CITY,MO
INAC515348145 100001UAL SYS ANAL DENVER, COLO
/*

```

Additional FILEDEVL Program Variables

<u>VARIABLE</u>	<u>MODE</u>	<u>DESCRIPTION</u>
BUFF1(41)	4I14	Spacer variable - used to facilitate disk reads and writes
TLBKLK	I4	Temporary storage variable for teaching level one backward link
TLFRLK	I4	Temporary storage variable for teaching level one forward link
MAJBLK	I4	Temporary storage variable for major area backward link
MAJFRL	I4	Temporary storage variable for major area forward link
LOCBLK	I4	Temporary storage variable for location preference one backward link
LOCFLK	I4	Temporary storage variable for location preference one forward link
BUF2(18)	18I2	Spacer variable - used to facilitate disk read and writes
INDOB(3)	3I2	Card input variable - date of birth
INDGYR	I2	Card input variable - degree and year awarded code
INYREX(3)	3I2	Card input variable - years of experience INYREX(1) - years of teaching experience INYREX(2) - years of educational administrative experience INYREX(3) - years of non-educational experience
INHSWF	I2	Card input variable - husband-wife teaching combination (1 - yes, 2 - no)
INMINS	I2	Card input variable - minimum salary acceptable--\$100's
INSEX	I2	Card input variable - sex code (1 - male, 2 - female)
INCTYS(5)	5A4	Card input variable - registrant's local city and state
INLPHN(2)	2A4	Card input variable - registrant's local phone number

INCOMP(4)	4I4	Card input variable - competencies
INSTYP(2)	2I4	Card input variable - institutions or organizations
INLOCP(3)	3I4	Card input variable - location preference codes
ISSN	I4	Card input variable - registrant's social security number
INCHAR	A4	Card input variable - card identification number
INCODE	I4	Card input variable - post graduation plans code
IRECNO	I4	Pointer to record number in TOTALREC file
INZIP	I4	Card input variable - registrant's local zip code
INMAR	I4	Card input variable - registrant's marital code
INLPRE	I4	Card input variable - location requirement (0 - no preference, 1 - a necessity, 2 - strong preference, 3 - preference)
ITEMP1	I4	Temporary storage variable
NTEMP1	I4	Temporary storage variable
MTEMP1	I4	Temporary storage variable

Appendix G.

Listing of CREATE Program.

```

// EXEC ASMOCLO
//ASM.SYSIN DD *
* CREATED PROGRAM BY RICHARD LONSINGER, AUGUST 1970.
* THIS PROGRAM ACCEPTS ONE DATA CARD WHICH HAS THE MAXIMUM
* GRADUATION DATE OF INTEREST PUNCHED IN MONTH/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 NOGEN
BALR 12,0
USING *,12
ST 13,SAVER+4
LA 13,SAVER
B BEGIN
SAVER DS 18F
BUFFIN DS CL480
BUFFOUT DS CL320
INPUT DS CL80
M1 DC F'-1'
*
* READ A CARD THAT CONTAINS THE MAXIMUM GRADUATION DATE FOR THE FILE
*
* BEGIN
OPEN (CARDIN,(INPUT)) OPEN CARD FILE
READ CARDEC8,SF,CARDIN,INPUT,S' READ MAX GRAD. DATE
CHECK CARDEC8
CLOSE CARDIN CLOSE CARD FILE
LA 5,BIGREC GET THE ADDR OF THE INDEX FILE BUFFER
L 4,FOUR40 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))
SR 6,6
LA 2,3399 SET UP 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 TAPEDEC8,SF,TAPEIN,BUFFIN,S' GET A RECORD
CHECK TAPEDEC8 CHECK FOR SUCCESSFUL READ
CLI 408(10),X'40' CHECK FOR BLANK GRAD DATE
BE REED GO READ ANOTHER REC IF IT IS
* CHECK FOR VALID GRADUATION DATE
CLI 408(10),X'F0'
BL REED
CLI 406(10),X'F0'
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 WASTFILE
      DS      DS      UD
      DS      D
      DS      F'0'
      DC      DC      X'40404040'
      DC      DC      X'000000CF'
      DC      DC      F'440'
      DC      DC      F'440'
      DC      DC      X'FOFO'
      DC      DC      X'FCFOFOCO'
      SPACE 3
      XC      TEMP(8),TEMP      ZERO TEMP
      CLI      9(10),X'40' CHECK FOR BLANK SSN
      BE      RECD DONT INCLUDE IF IT IS
      CLI      9(10),X'FO'
      BL      RECD
      CLI      1(10),X'FO'
      BL      RECD
      PACK      TEMP(8),1(9,10) CHANGE SSN TO
      CVB      AN 14 NUMBER
      ST      9,BUFFOUT PUT SSN IN OUTPUT BUFFER
      L      8,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(1,5) 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(8),TEMP
      CLI      371(10),X'40' CHECK FOR BLANK BIRTH MONTH
      DNE      *+10 OK
      MVC      370(2,10),HAFBLNK
      PACK      TEMP(8),370(2,10) CHANGE MONTH
      CVB      9,TEMP TO BINARY
      STH      9,BUFFOUT+24 STORE THE MONTH OF BIRTH
      CLI      373(10),X'40' CHECK FOR BLANK BIRTHDAY
      DNE      *+10
      MVC      372(2,10),HAFBLNK
      PACK      TEMP(8),372(2,10) 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),HAFBLNK
      PACK      TEMP(8),89(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
      RE      MOVNC
      CLI      34(10),X'FO'
      BNL      PACNC
      DC      32(3,10),FOFOFO
      MOVNC      MOVNC
      PACNC      PACNC
      PACK      TEMP(8),32(3,10) GET THE NUMERIC CURRICULUM
      CVB      9,TEMP
      STH      9,BUFFOUT+30 STORE THE NUMERIC CURRICULUM

```

```

MVC          BUFFOUT+32(4),35(10)  STORE THE ALPHA CURRICULUM
SR           9,9
IC           9,29(10)  GET THE SCHOOL CODE
N           9,0F      REMOVE THE ZONE
STH         9,BUFFOUT+36  STORE SCHOOL CODE
IC           9,30(10)  GET THE CLASS CODE
N           9,0F      REMOVE THE ZONE
STH         9,BUFFOUT+38  STORE THE CLASS CODE
IC           9,31(10)  GET RESIDENCE CODE
N           9,0F      REMOVE ZONE
STH         9,BUFFOUT+40  STORE RESIDENCE CODE
SR           9,9
CLI         429(10),X'E8'  DOES PLAN TO TEACH EQUAL Y?
BE          YES01
BL          NO01
LA           9,3        SET UNDECIDED = 3
STH         9,BUFFOUT+42  STORE CODE
B           NEXT01
LA           9,2        SET NO = 2
B           LABEL01
LA           9,1        SET YES = 1
B           LABEL01
SR           9,9
CLI         430(10),X'E8'  DOES PLAN GRAD SCHOOL EQUAL Y?
BE          YES02
BL          NO02
LA           9,3        SET UNDECIDED = 3
STH         9,BUFFOUT+44  STORE CODE
B           NEXT02
LA           9,2        SET NO = 2
B           LABEL02
LA           9,1        SET YES = 1
B           LABEL02
SR           9,9
XC          TEMP(8),TEMP
CLI         431(10),X'E8'  DOES REG AT PLACE CENTER EQ Y?
BE          YES03
LA           9,2        SET NO = 2
STH         9,BUFFOUT+46  STORE CODE
B           NEXT03
LA           9,1        SET YES = 1
B           LABEL03
SR           9,9
PACK        TEMP(8),405(2,10) GET THE MONTH OF GRADUATION
CVB         9,TEMP
STH         9,BUFFOUT+52
PACK        TEMP(8),407(2,10) GET THE YEAR OF GRADUATION
CVB         9,TEMP
STH         9,BUFFOUT+54
MVC         BUFFOUT+56(3),409(10) MOVE THE CITIZENSHIP
SR           9,9
IC           9,84(10)  GET THE VETERAN CODE
N           9,0F      REMOVE ZONE
STH         9,BUFFOUT+60
IC           9,85(10)  GET SEX/MAR CODE
N           9,0F      REMOVE ZONE
STH         9,BUFFOUT+62
MVC         BUFFOUT+64(7),238(10) GET LOCAL PHONE NUMB
MVC         BUFFOUT+72(18),97(10) GET LOCAL ADDR
MVC         BUFFOUT+92(18),118(10) GET CITY/STATE

```

```

CLI 140(10),X'40' CHECK FOR BLANK ZIP CODE
RE NEXTZIP
CLI 140(10),X'F0'
RL NEXTZIP
CLI 136(10),X'F0'
BE NEXTZIP
PACK TEMP(8),136(5,10) GET ZIP CODE
CVB 9,TEMP
ST 9,BUFFOUT+112
CLI 344(10),X'40' CHECK FOR BLANK ZIP
BE RITE
CLI 344(10),X'F0'
BL RITE
CLI 340(10),X'F0'
RL RITE
PACK TEMP(8),340(5,10) GET ZIP CODE
CVB 9,TEMP
ST 9,BUFFOUT+48 STORE PERM ADDR ZIP CODE
*
MVC BUFFOUT+116(20),301(10) GET PERMANENT ADDR
MVC BUFFOUT+136(18),322(10) GET CITY/STATE
MVC BUFFOUT+156(17),348(10) GET PERM RES PHONE NUMBER
SPACE 3
WRITE DFIDECB,SF,DISKFILE,BUFFOUT,'S' WRITE TOTAL RECORD
SPACE 2
CHECK DFIDECB CHECK FOR SUCCESSFUL WRITE
LA 5,8(5) INCR THE POINTER IN BIGREC
CR 8,4 CHECK FOR 440 IN BIGREC
BL REED GO READ ANOTHER TAPE RECORD
SPACE 2
WRITE DOUTDEC,SF,DISKOUT,BIGREC,'S'
SPACE 2
CHECK DOUTDEC
SPACE 2
BCTR 11,0 DCR BIGREC COUNT
SR 8,8
ST 8,COUNT ZERO COUNT
LA 5,BIGREC
B REED
SR 9,9
STH 9,4(,5)
WRITE DOUTDEC,SF,DISKOUT,BIGREC,'S'
SPACE 2
CHECK DOUTDEC
BCTR 11,0
* FORMAT THE REMAINDER OF TOTALREC AND INDEXREC
MVI BUFFOUT,X'00' ZERO OUT BUFFOUT
MVC BUFFOUT+112(56),BUFFOUT
MVC BUFFOUT+256(63),BUFFOUT
LCOPX1 WRITE TOTDEC,SF,DISKFILE,BUFFOUT,'S'
CHECK TOTDEC
BCT 2,LOOPX1
LOOPX2 WRITE INXDEC,SF,DISKOUT,BIFREC,'S'
CHECK INXDEC
BCT 11,LOOPX2
* END FORMATTING
SPACE 2
CLOSE TAPEIN,DISKFILE,DISKOUT
SPACE 2

```

```

L      13,SAVER+4
RETURN (14,12)
SPACE 2
CARDIN DCB DDNAME=SYSIN,DSORG=PS,RECFM=F,HLKSIZE=80,MACRF=R      X
TAPEIN DCB DDNAME=TAPEMAST,DSORG=PS,RECFM=F,BLKSIZE=480,MACRF=R,
          DEVD=TA,CEN=2,TRTCH=C,EODAD=TAPEND
DISKFILE DCB DDNAME=TCTALFIL,DSORG=PS,RECFM=F,BLKSIZE=320,MACRF=W
DISKOUT DCB DDNAME=DISKMAST,DSORG=PS,RECFM=F,BLKSIZE=3520,MACRF=W
BIGREC DS CL3520
END      CREATED
/*
//GO.SYSUDUMP DD SYSOUT=A
//GO.DISKMAST DD DSN=COCPI2PC.INDEXREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),SPACE=(TRK,(4)),DCB=(RECFM=F,BLKSIZE=3520)
//GO.TOTALFIL DD DSN=COCPI2PC.TOTALREC.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),SPACE=(TRK,(60,10)),
// DCB=(RECFM=F,BLKSIZE=320)
//GO.TAPEMAST 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 FORTSCLG
// FORT.SYSIN DD *
C DIRCREA PROGRAM BY RICHARD LONISINGER, AUGUST 1970.
C THIS PROGRAM SETS UP THE TWO FILES, CUMPOREN AND LOCROGDIR, 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 800
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 1001 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 6001 CARD (CC 1-4).
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*2 STATE, SECT
C INTEGER*4 TWENTY, FIFTY
C DIMENSION INPUT(25,8), MPUT(4), LPUT(50,4)
C DIMENSION INOUT(200), MYOUT(200)
C DEFINE FILE 30(40,800,L,K30), 31(11,800,L,K31)
C DATA IBLNK/,
C K30= 1
C K31= 1
C FIFTY = 0
C DO 10 I = 1,25
C DO 11 J = 1,4
11 INPUT(I,J) = IBLNK
C DO 10 J = 5,8
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
C 1 FORMAT(14,4A4)
C 2 FORMAT (212,A4)
C 5 FORMAT (' 121',I5)
C 6 FORMAT(' 154',I5)
C TWENTY = 0
C
C BEGIN BLOCKING AND OUTPUTTING OF COMPETENCIES, ORGANIZATIONS AND
C FUNCTIONS IN MULTIPLES OF 25 BY CODE NUMBERS.
C
C DO 20 I = 1,40
19 READ(1,1) NUMB, MPUT
C NUMB = NUMB - TWENTY
C IF (NUMB.GT. 25) GO TO 24
C DO 21 J = 1,4
21 INPUT(NUMB,J) = MPUT(J)
C GO TO 19
24 TWENTY = TWENTY + 25
C SET UP THE OUTPUT ARRAY AND WRITE ON DISK.
C
C ME = 0
C DO 121 LE = 1,25
C DO 121 KE = 1,8
C ME = ME + 1
121 INOUT(ME) = INPUT(LE,KE)
C WRITE(3,5) ME
C WRITE(30,K30) INOUT
C DO 27 IX = 1,25
C DO 27 JX = 1,4
27 INPUT(IX,JX) = IBLNK
C IF (1.EQ. 40) GO TO 20

```

```

NUMB = NUMB - 25
DO 29 JX = 1,4
29 INPUT(NUMB,JX) = MPUT(J)
20 CONTINUE
C END GROUP ONE.
C BEGIN BLOCKING AND OUTPUTTING 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) = IBLNK
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 SECTON 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 + 50
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(3,6) ME
WRITE(31,K31) MYOUT
DO 57 IX = 1,50
57 LPUT(IX,1) = IBLNK
IF (I .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.FT31F001 DD DSN=COCPL2PC.LOCRGDIR.REL,UNIT=SYSDA,VOL=SER=222222, X
// DISP=(OLD,KEEP),DCB=(RECFM=F,BLKSIZE=800)
//GO.FT30F001 DD DSN=COCPL2PC.COMPORFN.REL,UNIT=SYSDA,VOL=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.