A research report
submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
in the Department of Mathematical Sciences
Florida Technological University
June, 1976

389 423
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>PREFACE</td>
<td>v</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1. MANAGEMENT INFORMATION SYSTEMS: THEIR HISTORY AND A PROPOSED DEVELOPMENTAL APPROACH</td>
<td>1</td>
</tr>
<tr>
<td>2. SYSTEM IMPLEMENTATION</td>
<td></td>
</tr>
<tr>
<td>GENERAL INFORMATION ON DEMIS / SIS</td>
<td>9</td>
</tr>
<tr>
<td>SIS BACKGROUND</td>
<td>11</td>
</tr>
<tr>
<td>SYSTEM SPECIFICATIONS FOR SIS</td>
<td>12</td>
</tr>
<tr>
<td>SIS CONCEPT</td>
<td>18</td>
</tr>
<tr>
<td>SIS SYSTEM DESIGN</td>
<td>20</td>
</tr>
<tr>
<td>SIS FILE DESIGN</td>
<td>22</td>
</tr>
<tr>
<td>SIS Coding</td>
<td>24</td>
</tr>
<tr>
<td>3. USER PROCEDURES FOR SIS</td>
<td></td>
</tr>
<tr>
<td>GENERAL</td>
<td>26</td>
</tr>
<tr>
<td>MAINTAINING THE MATH-DATA-BASE</td>
<td>26</td>
</tr>
<tr>
<td>RUNNING SIS</td>
<td>29</td>
</tr>
<tr>
<td>CREATION OF STAT-HIST DATA BASE</td>
<td>35</td>
</tr>
<tr>
<td>CHANGING / DELETING DATA FROM THE STAT-HIST DATA BASE</td>
<td>37</td>
</tr>
<tr>
<td>REPORT FORMATS</td>
<td>37</td>
</tr>
<tr>
<td>ERROR MESSAGES</td>
<td>39</td>
</tr>
</tbody>
</table>
4. MAINTENANCE OF SIS
   GENERAL .................................................. 41
   SIS-PTU STUDENT RECORD INTERFACE .................. 42
   TIME PARAMETER ADJUSTMENTS .......................... 45
   HIPO / SOURCE LISTING CORRESPONDENCE .............. 45
   FILE MAINTENANCE ....................................... 46
5. FUTURE DEVELOPMENT OF DEMIS .......................... 47
   CONCLUSION ............................................... 49
BIBLIOGRAPHY ............................................... 51

APPENDIXES
   A. ERROR MESSAGES ....................................... 53
   B. CURRENT QUARTER TABLE ............................... 62
   C. SIS LISTING ........................................... 65
   D. HIPO INPUT-PROCESS-OUTPUT (I-P-O) DIAGRAMS ....... 66
   E. SIS DECK LISTING ...................................... 84
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>DEMIS Framework</td>
<td>10</td>
</tr>
<tr>
<td>2.</td>
<td>Student Roster by Major/Level of Study</td>
<td>13</td>
</tr>
<tr>
<td>3.</td>
<td>Student Roster by Faculty Advisor</td>
<td>14</td>
</tr>
<tr>
<td>4.</td>
<td>Student-Faculty Advisor Listing</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>Listing of Students Advised by Each Faculty Member</td>
<td>16</td>
</tr>
<tr>
<td>6.</td>
<td>Student Enrollment Report</td>
<td>17</td>
</tr>
<tr>
<td>7.</td>
<td>SIS Overview Diagram</td>
<td>19</td>
</tr>
<tr>
<td>8.</td>
<td>HIPO Visual Table of Contents (VTOC)</td>
<td>21</td>
</tr>
<tr>
<td>9.</td>
<td>SPIN-OFF Input From FTU Student Records</td>
<td>23</td>
</tr>
<tr>
<td>10.</td>
<td>MATH-DATA-BASE Card Input Formats</td>
<td>28</td>
</tr>
<tr>
<td>11.</td>
<td>Deck Format to Recreate MATH-DATA-BASE</td>
<td>31</td>
</tr>
<tr>
<td>12.</td>
<td>STAT-HIST Card Input Format</td>
<td>32</td>
</tr>
<tr>
<td>13.</td>
<td>SIS Job Deck Format</td>
<td>33</td>
</tr>
<tr>
<td>14.</td>
<td>Year-Quarter ('Y') Card Input Format</td>
<td>34</td>
</tr>
<tr>
<td>15.</td>
<td>Report Request Card Formats</td>
<td>36</td>
</tr>
<tr>
<td>16.</td>
<td>Deck Format to Build STAT-HIST-DB</td>
<td>38</td>
</tr>
<tr>
<td>17.</td>
<td>SPIN-OFF Test Data Card Input Formats</td>
<td>43</td>
</tr>
<tr>
<td>18.</td>
<td>Deck Format to Load SPIN-OFF With Test Data</td>
<td>44</td>
</tr>
</tbody>
</table>
The value of Computerized Management Information Systems (MIS) was first realized in the 1960's. Since that time numerous approaches have been developed to implement MIS. Some have failed, others have been developed as a backlash to these failures. This report traces, briefly, the history of computerized Management Information Systems, then proposes an approach for the development of a small MIS. This approach was evaluated through the development of the Department of Mathematical Sciences Information System (DEMIS) and its first operational Module, the Student Information System (SIS). The report is divided into five chapters. Chapter 1 traces the history of MIS and outlines the approach proposed for small system development. Chapter 2 discusses general information pertaining to SIS. Chapter 3 sets forth a complete yet simple set of operator instructions. Chapter 4 provides philosophy, approaches and requirements for system maintenance and modification, and Chapter 5 looks at the future of DEMIS and provides some concluding remarks.

Following the text is complete documentation on the system including the design package, a system listing, sample outputs, input card formats, configuration of maintenance and execution decks, a list of error messages with explanations and required corrective actions, and listings of file creation.
and SIS cataloging decks.

I gratefully acknowledge the assistance provided by the personnel of the FTU Computer Center whose help made completion of this project possible. I am especially indebted to William H. Branch, Thomas O. Peeples, James Radford, and Bernard L. Slessinger.
Chapter 1

MANAGEMENT INFORMATION SYSTEMS:
THEIR HISTORY AND A PROPOSED
DEVELOPMENTAL APPROACH

The use of the computer as a management tool is traceable back to the mid 1950's (Mathews, 8). It was first used to reduce clerical work, with applications expanding into payroll, personnel, inventory status, cost distribution, sales analysis and similar applications. The 1960's brought about a new role for the computer, that of providing information which could help in management decision making. In this regard, however, the numerous stand-alone applications which had been developed presented a problem. Each had its own data base which duplicated information found in the other data bases. Not only was this wasteful in terms of duplicate storage of data, but data consistency problems were noted when like data items from several different data bases were examined. While inconsistencies were frequently explainable (e.g. one data base was updated daily, while another might require only monthly updating) questions nevertheless arose as to why data was maintained in more than one location. The solution to these problems was seen as developing separate but integrated data bases which would solve problems of inconsistency and duplication of data, and provide a means of integrating data for use in management oriented information...
systems (MIS). The MIS required the variety of information which was then available. The software overhead required in the maintenance of integrated data bases soon led to the contemporary concepts of data base management systems which provide common data bases for a variety of applications. In the mid 1960's several factors combined to cause a tremendous increase in interest in Management Information Systems. These include the availability of integrated data bases, and the third generation computers with their increased speed and power. Management began to accept the computer and realized it's potential to provide information upon which management decisions could be made.

Unfortunately the late 1960's brought an era of unhappy experiences with MIS, many due to the "total" information concept which was not far enough along in its development to be placed into production. This concept also tended to drive an organization's information system rather than to augment and support existing (though perhaps non-automated) systems (Caruth 3:197)(the total concept is further discussed below). Management expected too much, too soon from their MIS, and the results were generally disappointing. The reaction to this experience was reevaluation of MIS theory and a tremendous proliferation of ideas in the literature.

Sherman C. Blumenthal (1) has written a classic text which classifies MIS approaches into six categories:
1. The **organizational** approach establishes independent systems along organizational lines, with interaction depending on higher level coordination.

2. The **data collection** approach is to collect, classify and store data for some unknown future use.

3. The **top down** approach holds that once the information needs of top management are determined, the system necessary to supply the information can also be determined.

4. The **data bank** approach sets up a pool of highly detailed, unclassified data for some undetermined future use.

5. The **integrated later** approach is, as the name implies, a non-plan, a philosophy of continuing to develop more independent applications and "integrate them later".

6. The **integrate now** (or "total systems") approach, provides an "instant MIS", integrating all ADP functions at once. This is the approach which led to the reevaluation of MIS in the late 1960's.

Blumenthal then presents his own philosophy, with which I am in general agreement. He supports a "bottom up" approach combined with a systems plan. The plan can be considered from two aspects; global, which is a "plan of projects" or a framework for classifying and integrating modules, and
local, which consists of project plans or modules. Using this format, the modules are developed based on priorities, specifications, and integration considerations established by the "framework". Modules are developed to support operating level management. Their "communication" with the data bases of other modules is assured by the specifications developed in the framework. Selected data produced by the modules is copied into other data bases for use by higher level management based on Blumenthal's concept that the information required by higher level management is a subset of that produced by the lower level modules.

Blumenthal also describes the formal activities which should be accomplished in developing a system. These include:

Proposal study,
Problem definition, priority, and budget consideration,
Formal written proposal,
Initial user system organization assessment,
Preparation of a feasibility study,
Presentation of the feasibility study to management,
Assignment of project responsibility, allocation of resources, delegation of a steering committee,
Project planning and control,
Development of functional requirements,
Designing of system specifications,
Programming and testing, and
Conversion and cutover.
The only exceptions to this formal procedure are those applications which are obvious minor development efforts such as "special one-time reports". Most organizations with a major data processing investment utilize the concept proposed by Blumenthal to some extent. Unfortunately, once formal procedures are established for development of a system, generally no system, no matter how small, can be developed without going through the established formal organizational procedures.

The preimplementation criteria are, in themselves, costly enough to eliminate all but major systems from consideration.

The result of this environment is that the required strong economic justification for the small application is impossible, and so small applications tend to either be dropped or to mushroom into large systems, resulting in major contracts and extended periods of time for implementation.

The purpose of this paper is to provide an alternative approach using Blumenthal's "Bottom Up" Theory, but based on the thesis that formal procedures are not always necessary (may be accomplished informally); that requirements do not have to be large enough to justify major investments of time and resources; and that an organization's information requirements can evolve in an open ended plan, based on existing information resources.

This project considers the management information re-
requirements for one department of a college located at a large university. The information requirements were very real, yet probably not economically justifiable as a large project. Previous attempts to design information systems for other colleges in the university had either failed or their planning had dragged on over an extended period of time without realization of an operational system. The length of time to implement the system, the inability to support or maintain it, or the lack of a systemic approach to building a system to meet all information requirements resulted in a general loss of interest and the failure of these systems to produce satisfactory results.

The system described by this project was developed with a different approach, which is characterized by informal, though systematic, development, and includes a generalized MIS framework, which is flexible and open ended, and the first module of the system. The scenario for this approach was:

1. The users were contacted to determine what immediate information requirements existed, and the general nature of possible future additions to the system.
2. Personnel of the computer center were contacted to determine the type of information that was available in existing data bases to support a system such as the users felt they needed.
3. The users were again contacted and their requirements reappraised to make the maximum use of available data, and then the minimum additional data requirements were agreed upon.

4. A major system was conceptualized which could contain a number of interrelated sub-systems.

5. The subsystem to support the immediate requirements was designed, coded, and implemented.

The system uses top down, structured, modular design and programming for ease of maintenance, and was designed so that it can be run using existing equipment and personnel resources.

User interest and enthusiasm in the project has been maintained at high level. This is attributable to several factors. The information supplied to the users was information they needed and in the required format. The time for development of the system was less than six months, during which the users received frequent, informal feedback on the status of the project. No elaborate justification for the project was required of the users, and rather than enlarging their requirements to justify the project, they were able to reduce requirements to the essential data that was needed. The result of the latter was that the system was able to operate using existing personnel and equipment resources which meant that the users did not have to divert
funds from other projects to pay for this system.

I feel that this concept of ADP support offers a responsive alternative to organizations which have MIS formal planning and design criteria, and where:

a) the information requirements are valid,
b) the information requirements are such that a large expenditure of time and money is not immediately justifiable, and
c) most of the required information is already available in existing data bases.
Chapter 2

SYSTEM IMPLEMENTATION

GENERAL INFORMATION ON DEMIS / SIS

In keeping with the approach described in Chapter 1, a flexible, open ended framework was developed for the Department of Mathematical Sciences Information System (DEMIS). This framework is graphically illustrated in figure 1. No detailed specifications were made for the proposed modules of DEMIS, as this would detract from the flexibility of the system. Generalized contents of the modules are, however, discussed in concept in Chapter 5.

The Student Information System (SIS) is the first module of DEMIS to be implemented, and is designed to provide information to support management and counselling activities of the Department. SIS has been constructed using contemporary concepts of modular, top down, structured design, and structured programming techniques. It has also been designed to require minimum operator time, training, and skill, as well as minimum maintenance.

SIS provides useful, required, up to date information for the Department of Mathematical Sciences. It establishes a flexible, expandable student data base and a student enrollment statistical data base to be used by DEMIS. It also provides an interface for receiving data elements from the FTU
With the continuing growth of student enrollment in the Department of Mathematical Sciences, it has become more and more apparent that certain required administrative information pertaining to student enrollment is not readily available, or, at least, not available in the required format. As a result, frequent time consuming searches through file cabinets containing student records, or through the numerous available computer printouts, is necessary to assemble the required data. Some of the more pressing information requirements are determining the number of students majoring in the various programs offered by the Department of Mathematical Sciences; determining the number and names of students who are in Post Baccalaureate Status so their progress toward achieving provisional or regular status may be periodically reviewed; providing faculty advisors with immediate background data on students whom they advise as well as data concerning their overall progress. Often, various faculty advisor listings must also be manually modified and updated.

These problems indicated a need for an information system within the Department. DEMIS was developed to supply this information.
SYSTEM SPECIFICATIONS FOR SIS

In order to meet the requirements stated above, the following specifications were developed of SIS (the first module of DEMIS).

1. Outputs of the system will include:
   A. A roster of students currently enrolled ordered by major, level of study (undergraduate, post baccalaureate, or graduate) and alphabetical by name (figure 2).
   B. A roster of students ordered by faculty advisor (figure 3).
   C. Listing of student-faculty advisor assignments (figure 4).
   D. Listing of the students advised by each faculty member (figure 5).
   E. A statistical printout which shows student enrollment at quarter intervals (figure 6).

2. The system should be designed so that it can be operated by clerical personnel in the department.

3. Complete documentation is required for maintenance and modification and to facilitate the addition of other modules to DEMIS.

4. Data maintained in a Mathematics Department Data Base should be minimal, with the greatest portion
STUDENT HOSTEL AS MAJOR / LEVEL OF STUDY

PAGE 2
<table>
<thead>
<tr>
<th>Name</th>
<th>Local TEL</th>
<th>Sex</th>
<th>Marital Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shu Mei</td>
<td>664 717 6-2457</td>
<td>M</td>
<td>Single</td>
</tr>
<tr>
<td>Jun</td>
<td>554 667 6-2457</td>
<td>M</td>
<td>Single</td>
</tr>
<tr>
<td>Li</td>
<td>325 224 6-2457</td>
<td>F</td>
<td>Single</td>
</tr>
</tbody>
</table>

Address

1010 University Street
Chicago, IL 60608

Phone: 312-637-4473
Fax: 312-637-4474

May 26, 1976

Faculty Advisor of Student Poster
Department of Mathematical Sciences
### DEPARTMENT OF MATHEMATICAL SCIENCES

**FACULTY ADVISOR LISTING**

**GRADUATE MATHEMATICAL SCIENCE MAJORS**

**MAY 26 1976**

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>FACULTY ADVISOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>JONES</td>
<td>JOHN P</td>
</tr>
<tr>
<td>JORGENSEN</td>
<td>OSCAR</td>
</tr>
<tr>
<td>WASHINGTON</td>
<td>GEORGE C</td>
</tr>
</tbody>
</table>

**FIGURE 4**

**STUDENT-FACULTY ADVISOR LISTING**
THE FOLLOWING NAMED STUDENTS ARE ADVISEES OF DR. A. C. BROWN

BAXTER  JIMMY BUD
CONROY  MARY ELLEN
LEU  LEE SHU
SHOEMACHER  WILLIAM R.

FIGURE 5
LISTING OF STUDENTS ADVISED BY EACH FACULTY MEMBER
### DEPARTMENT OF MATHEMATICAL SCIENCES

**STUDENT ENROLLMENT**

**MAY 26, 1976**

#### FIGURE 6

<table>
<thead>
<tr>
<th>Course</th>
<th>COMP</th>
<th>MATH</th>
<th>STAT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>711</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNDERGRADUATE</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>2</td>
<td>3</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>POST BAC</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td><strong>704</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNDERGRADUATE</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>2</td>
<td>3</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>POST BAC</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td><strong>703</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNDERGRADUATE</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>POST BAC</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td><strong>694</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNDERGRADUATE</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>POST BAC</td>
<td>2</td>
<td>3</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td><strong>692</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNDERGRADUATE</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>GRADUATE</td>
<td>3</td>
<td>3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>POST BAC</td>
<td>1</td>
<td>3</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>16</td>
</tr>
</tbody>
</table>
of required data extracted from the FTU Student Records.

5. The system must operate on the CFRDC facilities.

SIS CONCEPT

SIS can be described in terms of three general (somewhat simplified) processes as shown in figure 7. These are: update the MATH-DATA-BASE file; load the SPIN-OFF file; and generate reports.

The MATH-DATA-BASE file is one of two data bases maintained by the Department of Mathematical Sciences (the other is the STAT-HIST file which is discussed below). This data base contains four data fields for each student record; faculty advisor name and SSN (Social Security Number), student attendance/status data, student association membership data, and the student SSN which is used to key the student MATH-DATA-BASE record to the FTU Student Record. Additions, changes, and deletions to this data base are made using punched cards. The latter are read into a temporary MATH-CHANGE file where they are sorted and then merged with the MATH-DATA-BASE into a temporary MATH-REORG file. This is then read back into the MATH-DATA-BASE.

The temporary SPIN-OFF file is loaded with selected data fields from FTU records of all students with majors of COMP,
MATH, or STAT (an alternate method for loading SPIN-OFF with test data is also provided). Whenever SIS is run, it retrieves the SPIN-OFF file records and uses the data to construct a temporary STUD-DATA-BASE file.

When a report is requested, the records from the MATH-DATA-BASE are merged into the STUD-DATA-BASE, which is then sorted on various combinations of keys to provide the requested reports. Whenever a statistics report is requested, a new set of current student enrollment statistics is compiled and added to the STAT-HIST file before the report is generated.

SIS SYSTEM DESIGN

The System design was developed and implemented through use of IBM's Hierarchy plus Input Process Output (HIPO) technique for system design. This involves the use of system specifications (already discussed), a hierarchical Visual Table of Contents (VTOC) (figure 8), and corresponding Input-Process-Output diagrams (appendix D).

The result of this design process is a modular, structured system design, as indicated in the VTOC. Each module is "called" by its higher module and, in turn, calls its lower modules. This approach has greatly simplified the design process. It also facilitates maintenance and changes because the appropriate module can be quickly located, and usually any changes made to it will only affect its "parent" and
Department of Mathematics Information System DEMIS 1.0

Provide student information SIS 2.0

Maintain Math Dept data base MATH-LOAD-UPDATE 2.1

Create new Math Dept record CREATE-MATH-REC 2.1.1

Update Math Dept data base record UPDATE-MATH-REC 2.1.2

Maintain statistics history file STAT-HIST 2.1.3

Load FTU student data LOAD-FTU-DATA 2.2

Capture FTU data on spin-off file GET-SPIN-OFF 2.2.1

Manually load FTU data on spin-off file LOAD-SPIN-OFF 2.2.2

Load STUD-DATA-BASE from SPIN-OFF LOAD-STUD-DB 2.2.3

Generate student reports GENERATE-REPORTS 2.

Merge FTU and data MERGE-DATA 2

Compute and print student statistics PRINT-STAT 2

Print student reports PRINT-REPT 2

FIGURE 8
HIPO VISUAL TABLE OF CONTENTS (VTOC)
Other DEMIS subsystems, when added, are numbered 3.0, 4.0, 5.0...

Common subroutines

Print subroutine
PRINT-RTN 4,1

Generate student reports
GERRATE-REPORTS 2.3

Merge FTU and math data
MERGE-DATA 2.3.1

Compute and print student state
PRINT-STAT 2.3.2

Print student reports
PRINT-REPT 2.3.3

LEGEND

Data Movement
Control Flow
Data Reference
Pointer

Subroutine is executed and control returns to next step.
"offspring" modules.

The Input-Process-Output diagrams, one of which is provided for, and keyed to, each module in the VTOC, visually and narratively describe the logic, the required inputs, and the outputs produced by each module. The coding was produced so that each module is implemented in a "section" of code which can quickly be identified for maintenance purposes.

Every attempt has been made to use corresponding terms in the VTOC, the Input-Process-Output diagrams, and in the coding.

Close coordination has been made with personnel at the FTU Computer Center, to ensure the compatibility of DEMIS with the FTU Student Records System and to facilitate the use of student record data by DEMIS.

SIS FILE DESIGN

The files used by SIS include two permanent files, five temporary files, and a variety of utility sort files. The two permanent files were established to store Mathematics Department unique data (MATH-DEPT-STUD-DB) and student enrollment data (STAT-HIST-DB).

Prior to running SIS a temporary file, SPIN-OFF, must be loaded with certain data from the FTU Student Records (figure 9). This data is placed in a temporary STUD-DATA-BASE file. The MATH-DEPT-STUD-DB is then merged into the
2LAST-NAME PIC X(18).
2FRST-MDLE-NAME PIC X(30).
2SSN PIC 9(09).
2APPL-TYPE PIC 9(01).

2MAJOR-1 PIC 9(04).
2PTU-SUMMARY.
  2PTU-LAST-QTR PIC 9(03).
  2PTU-GPA PIC 9999.
2SEX PIC X(01).
2INDX PIC X(200).

* 2GRD-IND PIC X(01).
* 2START-SUB PIC X(02).
* 2END-SUB PIC X(02).
2HOME-STREET PIC X(20).
2HOME-CITY PIC X(20).
2HOME-STATE PIC X(02).
2HOME-ZIP PIC 9(05).
2HOME-PHONE PIC X(10).
2KIN-STREET PIC X(20).
2KIN-CITY PIC X(20).
2KIN-STATE PIC X(02).
2KIN-ZIP PIC 9(05).
2MARITAL-STATUS PIC 9(01).

NOTE: DATA REQUIRED FOR ALL STUDENTS EVER ENROLLED IN FTU
WITH MAJOR CODE OF 0701, 1701, 1702, OR 1790

FIGURE 9
SPIN-OFF INPUT FROM FTU STUDENT RECORDS
STUD-DATA-BASE, to provide for generation of the required reports. Each time new enrollment statistics are generated, they are added to the STAT-HIST-DB to provide a continuous record of enrollment data.

All files used by SIS are sequential. The decision to use sequential files was made based on consideration of the following factors:

a) it is anticipated that large volumes of data will be processed,
b) data from the FTU Student Record Data Base will be sorted in sequence of SSN,
c) conversion from disk to tape, if required, will be facilitated,
d) data must be repeatedly sorted on different keys to produce the required outputs,
e) MATH-DEPT-STUD-DB is stored sequentially by SSN, so that change data can be placed on a file, sorted on SSN, then merged with the MATH-DEPT-STUD-DB, and
f) numerous and frequent changes to the SIS data bases are anticipated which makes other types of files (e.g. ISAM) inefficient.

SIS CODING

DEMIS was coded using COBOL, primarily to facilitate
interfacing with the FTU Student Record System which also uses COBOL. The coding has followed the logic of the HIPO system design, but, of course, is in more detail and includes numerous error checking routines (appendix A) which are not reflected in the HIPO design package. In other cases, where some of the higher level language characteristics of COBOL result in instructions which accomplish a variety of actions, the HIPO diagrams might be more detailed than the coding.

Every effort has been made to use the self-documenting characteristics of COBOL. Storage areas, variables, file names, paragraph names, and section names are written in a descriptive way, providing hopefully meaningful mnemonic names.

Every effort has also been made to provide easy to understand logic flow throughout the program. This, coupled with the program modularity, should facilitate reading of the code. Wherever possible, coding has been designed for clarity, even at the expense of efficiency. Package routines and other features are used wherever possible (e.g. COBOL sort-merge feature and string option).
Chapter 3

USER PROCEDURES FOR SIS

GENERAL

SIS has been designed to require minimum operator training and involvement. It provides a relatively foolproof system providing the procedures listed below are followed.

Many errors in keypunching and out of sequence card input will be detected by SIS and a warning message provided. Some errors, however, cannot be detected by the system (e.g. SSN or spelling of name, address, etc.). Critical data should be double checked by the keypunch operator. Of particular importance is SSN because it is used as the key to locate and update records in most of the files.

All reports can be obtained without complete MATH-DEPT-STUD-DB records. Report formats which are keyed to advisor SSN will, however, be affected by the absence of faculty advisor data (this is further explained below).

MAINTAINING THE MATH-DATA-BASE

The primary task of operating SIS is the maintenance of the MATH-DEPT-STUD-DB. All other data is automatically computed (i.e. statistics) or is provided from the FTU Student Record Data Base.

Changes to the MATH-DEPT-STUD-DB refer to introducing
data into a student record, changing or deleting data in a record, or deleting an entire student record. These changes are accomplished by key punching and inputting cards in the appropriate format (figure 10). When this data is entered into the computer, a search will be made using SSN as a key, to see if a record exists for that particular student. If one does exist, it will be updated. If no record exists, a new record will be created using whatever data is submitted.

There are three data items, in addition to SSN, which must be maintained for each student: faculty advisor (name and SSN), quarters attended FTU and status, and association memberships. The latter two items are for information only (i.e., not used as sort keys) and their absence will in no way affect the operation of SIS (other than blank fields on the printouts). Incomplete faculty advisor data, however, will change the output formats because several reports are keyed to the advisor SSN. All students without an assigned faculty advisor are grouped together as having no assigned advisor.

Building the MATH-DEPT-STUD-DB will be a time consuming process and it is recommended that the data be phased in starting with faculty advisor input.

Sequencing of MATH-DEPT-STUD-DB card changes is immaterial. They are sorted by SIS before any records are updated. There is no need, for example, to place two changes pertaining to the same student physically together in the group of MATH-DEPT-STUD-DB changes.
<table>
<thead>
<tr>
<th>ADVISOR INPUT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT SSN</td>
<td>ADVISOR NAME</td>
</tr>
<tr>
<td>99999999999999999999999999999999</td>
<td>9999999999999999999999</td>
</tr>
<tr>
<td>F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80</td>
<td></td>
</tr>
</tbody>
</table>

| QUARTERS ATTENDED PTU AND STATUS (AS MANY ENTRIES AS ARE TO BE ADDED) |
| STUDENT SSN                       | STATUS               |
| 99999999999999999999999999999999 | 9999999999999999999999 |
| F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80 |

| DELETE "QUARTERS ATTENDED PTU AND STATUS" ENTRY |
| STUDENT SSN                       | ENTRY TO BE DELETED |
| 99999999999999999999999999999999 | 9999999999999999999999 |
| F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80 |

| ASSOCIATION MEMBERSHIPS            | NOTE: NO ENTRY MUST EXCEED 10 CHARACTERS, |
| STUDENT SSN                       | ENTER MEMBERSHIPS, SEPARATING EACH ONE BY A COMMA |
| 99999999999999999999999999999999 | 9999999999999999999999 |
| F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80 |

| DELETE ASSOCIATION MEMBERSHIP      | |
| STUDENT SSN                       | ENTER MEMBERSHIP TO BE DELETED |
| 99999999999999999999999999999999 | 9999999999999999999999 |
| F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80 |

| DELETE RECORD FROM MATH DATA BASE  | |
| STUDENT SSN                       | |
| 99999999999999999999999999999999 | 9999999999999999999999 |
| F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19 F20 F21 F22 F23 F24 F25 F26 F27 F28 F29 F30 F31 F32 F33 F34 F35 F36 F37 F38 F39 F40 F41 F42 F43 F44 F45 F46 F47 F48 F49 F50 F51 F52 F53 F54 F55 F56 F57 F58 F59 F60 F61 F62 F63 F64 F65 F66 F67 F68 F69 F70 F71 F72 F73 F74 F75 F76 F77 F78 F79 F80 |
RUNNING SIS

The following conditions must be met before output can be obtained from SIS.

1. The SPIN-OFF file must be created from the FTU Student Records.
2. The MATH-DEPT-STUD-DB file must have been created.
3. The STAT-HIST file must have been created.
4. SIS must be catalogued.
5. The execution deck must be loaded.

The SPIN-OFF file is created by the FTU Computer Center. This should be accomplished as soon as possible after the start of each academic quarter, once the INDEX field of all FTU student records has been updated. It will normally require several days notice for personnel at the center to create this file.

The MATH-DEPT-STUD-DB file is permanently stored. However, should the file be destroyed, it must be regenerated before SIS will work. To accomplish this, load DECK #3 (appendix E) into the card reader and verify the creation and cataloging of the file from the printed listing. This will recreate the MATH-DATA-BASE file without student data. Reloading student data is accomplished by rerunning all previously run changes to the data base, in the same sequence in which they were originally run, with each data deck separated
by a card with an '‡' in column #1 (see figure 11). The effect of this procedure is to recreate the database exactly as it had been previously built. The old change cards must, therefore, be saved and kept in a secure location. They form a data backup for use in case the MATH-DEPT-STUD-DB is damaged or destroyed.

The STAT-HIST file, like the MATH-DEPT-STUD-DB, is a permanent file. If it should be destroyed DECK #4 (appendix E) must be loaded to recreate the file (without data). To reload the data into the file there are two alternate procedures. The first is to recreate the history data as discussed in "Creation of STAT-HIST Data Base" below. The second is to take the most recent printout and use the information there to complete an 'H' card (figure 12) for each quarter in which data is available. The latter procedure is probably the best approach, and will provide a permanent backup deck. The backup deck may then be loaded with the next SIS run.

SIS is permanently catalogued on the CFRDC system, however should it be destroyed, it may be recatalogued by loading DECK #2 (appendix C).

The execution deck consists of several items (figure 13), which must be in proper sequence. DECK #1 contains the instructions which call SIS to execute the program. This must be immediately followed by a 'Y' card (figure 14). This
FIGURE 11
DECK FORMAT TO RECREATE
MATH-DATA-BASE
**ALTER STAT - HIST DATABASE - 'H' FORMAT**

<table>
<thead>
<tr>
<th></th>
<th>COMP-UG</th>
<th>COMP-GR</th>
<th>COMP-PE</th>
<th>MATH-UG</th>
<th>MATH-GR</th>
<th>MATH-PE</th>
<th>STAT-UG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

**NOTES:**
1. COLUMNS 2 THROUGH 32 MUST EACH CONTAIN A DIGIT 0 THROUGH 9.
2. CURRENT QTR CODE IS THE QUARTER FOR WHICH THE STATISTICAL DATA IS TO BE CHANGED. USE DATA FOUND IN APPENDIX B.

**FIGURE 12**

**STAT-HIST CARD INPUT FORMAT**

<table>
<thead>
<tr>
<th></th>
<th>COMP-UG</th>
<th>COMP-GR</th>
<th>COMP-PE</th>
<th>MATH-UG</th>
<th>MATH-GR</th>
<th>MATH-PE</th>
<th>STAT-UG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
Figure 13
SIS Job Deck Format

- Deck #1
- Math-DBase Changes (Optional)
- *
- Report Requests and 'H' Cards (Optional)
YEAR AND QUARTER - 'Y' FORMAT

CURRENT QUARTER ENTRY FROM APPENDIX B

FIGURE 14

YEAR - QUARTER 'Y' CARD INPUT FORMAT
card supplies SIS with the present academic year and quarter. Based on this information a search of student records pertaining to that quarter is made. Failure to enter a 'Y' card will cause an immediate termination of the program. Use of incorrect data on this card will cause the wrong data to be searched in the student records, with undeterminable results. The 'Y' card contains four characters. The first character is the letter 'Y'. This is followed by three numbers representing the current fiscal quarter (appendix B). A card with an '*' in column 1 must follow the 'Y' card. Failure to enter an '*' will cause the first MATH-DEPT-STUD-DB change card to be ignored, and a message to this effect will be printed (appendix A). It may also cause damage to the SPIN-OFF file. The next items in the deck are the MATH-DEPT-STUD-DB changes. They may be in any sequence. The system will sort them prior to posting changes to the MATH-DEPT-STUD-DB records (changes are optional when SIS runs). An '*' must follow the changes. Report requests (figure 15) follow, and may be run in any sequence (report requests are optional).

CREATION OF STAT-HIST DATA BASE

The 'Y' card, as mentioned above, indicates to SIS which quarter is to be searched in the SPIN-OFF file (FTU Student Data) to determine current enrollment and statistical data.
To access data from previous quarters, the 'Y' card must indicate the quarter for which data is to be collected. Therefore, to build the STAT-HIST-DB file, successive runs of SIS must be made, starting with the first quarter (691) up to the current quarter. The quarters are listed in appendix B. This data is automatically stored in the STAT-HIST-DB file and printed with each 'C' report request.

The data deck required to build the STAT-HIST-DB is shown in figure 16.

CHANGING / DELETING DATA FROM THE STAT-HIST DATA BASE

Statistical data stored in the STAT-HIST-DB file may be altered or deleted by using an 'H' format card (figure 12). To add new data, complete the card, as shown in figure 12, with an 'H' in column 1 and numeric data (0-9) completely filling columns 2 through 32. Failure to enter data as required will cause the 'H' card to be disregarded by SIS. To delete an entry from STAT-HIST-DB file, enter the date which is to be deleted, then fill columns 8 through 32 with 'Ø's.

REPORT FORMATS

Report formats are shown in figures 2 through 6. The student enrollment report (figure 6) will display all history data in the STAT-HIST-DB file, with the most recent
NOTE: 'Y' CARD MUST BE CHANGED ON EACH SUCCESSIVE RUN. THE 'Y' CARD DETERMINES THE QUARTER FOR WHICH DATA WILL BE COMPUTED AND ENTERED INTO STAT-HIST-DB.

FIGURE 16
DECK FORMAT TO BUILD
STAT-HIST-DB
statistics displayed first. A new entry will be made to the STAT-HIST-DB file every time a 'C' report (figure 15) is requested. If an entry already exists in the file for the quarter indicated on the 'Y' card, the new data which is generated will update the existing entry.

The 'R' report produces two sets of reports (figure 2 and 3): a roster of students currently enrolled by major and level of study, and the same data grouped by faculty advisor.

The 'A' report (figure 5) provides one listing for each faculty advisor showing the students he advises. Those students who are not assigned an advisor will appear on one listing, and new / temporary advisor names may be manually entered on the printout.

The 'F' report (figure 4) produces a student-faculty advisor listing by major and level of study.

ERROR MESSAGES

SIS incorporates a number of error checking routines. When an error is detected, a message is printed out to advise the user of the problem, what action SIS has taken, and in some instances what further action the user should take. Once the error message is printed, SIS continues to execute if at all possible. It is, therefore, essential that each output produced by SIS be reviewed and, if error messages occurred,
corrective action should be taken by the user. To this end a complete list of SIS generated messages, along with appropriate user responses, appears in appendix A.
Chapter 4

MAINTENANCE OF SIS

GENERAL

Problems arising from SIS operation should be easy to isolate and correct using the documentation provided with this package. This documentation includes the HIPO package (figure 8, appendix D) and a listing of SIS. Other figures found in this report should also be of use in the maintenance of SIS.

SIS has been designed in a modular fashion in the anticipation that additional DEMIS modules will be designed to interact with SIS to perform other functions. Expansion of the MATH-DEPT-STUD-DB, provided that it is accompanied by an identical expansion of the associated sort file, will not affect the operation of SIS. All data transfers, to and from the data base, use the COBOL MOVE CORRESPONDING option. This option has been used so that the locations of the corresponding fields within the sending and receiving records is not critical. This principle holds true for modifications in the SPIN-OFF data received from the PTU Student Records.

For testing purposes, SIS incorporates a procedure for loading the SPIN-OFF file using card input test data. This data is designed to be input in card form using the format
described in figure 17. This data is provided as DECK # 5 for future use and when used should follow the input sequence shown in figure 18. Should additional FTU Student Record data be required by SIS or new DEMIS subsystems, it may be simulated, for test purposes, by defining additional card input formats, adding additional fields to the description of the SPIN-OFF file and its associated sort file, and making appropriate changes to the LOAD-FTU-DATA section.

Should additional MATH-DEPT-STUD-DB fields be required, new card input formats must be defined and additional fields added to the MATH-DEPT-STUD-DB, its associated sort file, and the MATH-CHANGE file. The program sections which process the MATH-DEPT-STUD-DB must also be appropriately modified.

SIS - FTU STUDENT RECORD INTERFACE

A SPIN-OFF file has been temporarily created to contain extracts of the FTU Student Records on all students whose major is COMP, MATH, or STAT. Once this data has been loaded into the SPIN-OFF file by the personnel of the FTU Computer Center, it is available for use by SIS.

The format and data required for the SPIN-OFF file is as shown on the SIS listing (appendix C) as the SPIN-OFF-PD data record. The fields of this record are formatted exactly as the corresponding fields in the FTU Student Record Data Base except for a '2' prefix for each field descrip-
**BASIC DATA**

<table>
<thead>
<tr>
<th>SSN</th>
<th>LAST NAME</th>
<th>FIRST, MIDDLE, MAIDEN NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
</tr>
</tbody>
</table>

**STUDENT ADDRESS**

<table>
<thead>
<tr>
<th>SSN</th>
<th>STREET</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
<th>PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
</tr>
</tbody>
</table>

**PARENTS ADDRESS**

<table>
<thead>
<tr>
<th>SSN</th>
<th>STREET</th>
<th>CITY</th>
<th>STATE</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
</tr>
</tbody>
</table>

**INDEX FIELD**

<table>
<thead>
<tr>
<th>SSN</th>
<th>QTR 691</th>
<th>QTR 692</th>
<th>QTR 693</th>
<th><strong>NOTES:</strong> 1. FOR QUARTERS IN WHICH STUDENT WAS NOT ENROLLED ENTER '*****'. 2. FOR QUARTERS IN WHICH STUDENT WAS ENROLLED CONSTRUCT CODE AS FOLLOWS: 1st CHAR (1 or 2 IS GRADUATED), 2d and 3d CHAR (STARTING CR3), 4th and 5th CHAR (ENDING CR3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
<td>999999999</td>
</tr>
</tbody>
</table>

**NOTES:** 1. TEST DATA CARDS MUST BE HAND SORTED. 2. TEST DATA FOR SIS IS DESIGNED TO BE USED WITH A 'Y' CARD WITH 'Y711'.

**FIGURE 17**

**SPIN-OFF TEST DATA CARD INPUT FORMATS**

<table>
<thead>
<tr>
<th>SSN</th>
<th>999999999</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999999</td>
<td>999999999</td>
</tr>
</tbody>
</table>

---

**IBM**

**INFORMATION RECORDS DIVISION**

**MULTIPLE-CARD LAYOUT FORM**

---

**Company**

**Application**

**Company**

**Date**

**Job No.**

**Sheet No.**
tion on the FTU Student Record format (e.g. 'LAST-NAME' of the SPIN-OFF-REC of SIS is the receiving field of '2LAST-NAME' of the FTU Student Record). The data description for the required FTU Student Record input is shown in figure 9. The picture description and length of all SPIN-OFF fields are identical to the corresponding record formats of the FTU Student Record Data Base.

TIME PARAMETER ADJUSTMENTS

Periodic reviews should be made of the time limit parameters assigned to SIS (both CPU and I-O). Should the actual run times exceed the present limits, appropriate adjustments must be made to the job card in DECK #1.

HIPO / SOURCE LISTING CORRESPONDENCE

The following procedure may be used to help locate a specific area of SIS where a program modification is to be made:

1) locate the area in question on the HIPO VTOC,
2) refer to the I-P-O diagram whose number is referenced by the VTOC,
3) when the appropriate I-P-O diagram is found, locate the reference note in the source listing for that particular I-P-O diagram, and
4) read the code to locate the appropriate coding to be modified.

The source program for SIS is available at two locations. The original SIS deck is stored in the Department of Mathematical Sciences office. The source listing is also recorded on magnetic tape in the CFRDC PANVALET system.

FILE MAINTENANCE

Periodically, SIS should be reviewed to determine if the file space allocation requires modification. Initially SIS was created with the minimum allocation of "SPACE=(TRK,(1,1))". This limit should cause no problems in executing SIS. As certain data bases increase in size, however, inefficiencies may develop and the space parameters should be adjusted.

The sort file work areas used by SIS have space parameters of "SPACE=(TRK(20),,CONTIG)", which should be adjusted upward if the sort merge routines develop problems or if they appear to be inefficient due to an increase in the length of the data files. The first indication of the space problem in sorting will be encountered during a sort-merge of the STUD-DATA-BASE. Adjustments of these space parameters require changes to JCL cards for SORTWK01, SORTWK02, and SORTWK03.
Chapter 5

FUTURE DEVELOPMENT OF DEMIS

The following is a very generalized "framework" for the future development of DEMIS, and in keeping with the basic concepts of this paper, the framework should remain flexible and open ended. Possible additional modules of DEMIS are shown in figure 1 and discussed below. Naturally, before undertaking the development of a new module for DEMIS, the designer should determine the system requirements and specifications.

A Student Counselling Module for DEMIS would offer an increased scope of the counselling feature offered by SIS. The counselling module could provide a degree plan for students based on their areas of interest, program requirements, transfer credits and other appropriate considerations. It should produce a written program of instruction for each student, with a copy for his faculty advisor.

A Student Scheduling Module might receive input based on data made available by the Student Counselling Module, a student's program requirements, the work load each student desires to carry, and faculty availability. Projected offerings and a recommended schedule for each student for the following two or more quarters could then be made. Such a system would help optimize class sizes, provide an orderly sequence of course offerings for students, bring about optimum faculty...
utilization, reduce faculty workloads and could possibly result in a tangible monetary savings to both students and the school. The facilities for computer science majors (e.g. computer facilities) could be better utilized. Certain programming courses might, for example, be concentrated during summer quarters when school attendance is low. During the remainder of the year a more balanced schedule of programming and non-programming courses could be offered.

The FTU Alumni Association is currently investigating the development of a new computerized alumni information system. If these plans are realized, the data bases developed to support that system could be made available to DEMIS. Some of the information which might be of value would be a periodic listing of alumni and the type of jobs they hold. This would be valuable for curriculum development and counselling and could provide feedback to faculty so that adjustments might be made to the approach and content of individual course offerings. The result could be dynamic, "self evaluating" curriculum development procedure for the Department.

A Departmental Statistical Summary Module may be designed to provide statistical data, primarily from other DEMIS modules and data bases, for management use within the department. Such data might include student enrollment, faculty qualifications and specialties, alumni employment, evaluation of the "strengths" of FTU students (based on course balance),
average grade evaluation, and other similar areas. This data could be used to support management decisions, to provide information to potential students, and to determine qualifications and prerequisites for admission based on the profile of a typical successful student. The information provided by this system might also provide a profile to be used in the selection of new faculty members.

These are some modules which may be added to DEMIS to increase the management type information available to the staff and faculty of the Department of Mathematical Sciences. In addition to supporting activities in the Department, several other benefits can accrue from development of DEMIS. The design and construction of additional modules as student projects can provide an excellent opportunity for students interested in computer business applications, and specifically management information systems. The experience gained in developing a production system cannot be matched in any other way and can integrate the formal course work of the student. It could better prepare the student for the business world by providing him with experience and confidence.

Other student projects might be realized in generalizing DEMIS so that it can be used in other departments.

CONCLUSION

SIS provides student information and is the initial module
of DEMIS. SIS is designed to be operated with a minimum of
time, effort, and technical expertise, and it provides use-
ful data to support administrative and student advisement
activities within the Department of Mathematical Sciences.

The approach used to produce DEMIS is a valid concept in
MIS development. While the case study presented in this paper
pertained to an educational institution, it is equally applica-
ble to an organization with existing ADP facilities and
sufficient capacity to support an additional MIS.

The generalized flexible framework forces consideration
of the other potential modules without the requirement for a
formal study and the danger of delay in satisfying immediate
information requirements. It also prevents the danger of be-
coming "locked" into a rigid framework which will not provide
the flexibility necessary to meet rapidly changing information
requirements.

The implementation of individual modules creates a quick-
ly realizable goal, and can produce tangible results in a
short period of time. This makes this approach both attrac-
tive and economical.

The informal procedures to implement modules can, how-
ever, be dangerous. Certainly any large system which repre-
sents a sizeable and costly undertaking should meet formal
procedural requirements as listed in Chapter 1.
Bibliography


Appendix A

ERROR MESSAGES
ERROR MESSAGES

The error messages generated by SIS are listed below in alphabetical order with an appropriate user response. The more significant error messages generated will include an '*' as the first character of the error message. Note that where the message includes the comment 'THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)' that the referenced card has been completely ignored by SIS.

01 MSG: ATTENDANCE DATA ENTRY TO BE DELETED NOT FOUND. THE FOLLOWING INPUT CARD DISREGARDED: (CARD IMAGE)

User Response: A request was made to delete an attendance data entry, and the entry could not be found. Check for the correct SSN on input cards, check to make sure that column 1 has the correct code, and check previous listing of data to ensure that the entry to be deleted is exactly the same as on the listing. The listing should also be checked to ensure that previously entered incorrect data is not on the record (i.e. an inbeded space in the data string, a leading space in the data string, or a missing or extra character in the data string). If the input card is incorrect make appropriate changes. If an incorrect data string is found in
the student record the entire data field must be purged using a 'Z' format card followed on the next SIS run with a 'Q' format card with corrected attendance record data.

02 MSG: * CANNOT DETERMINE IF (STUDENT LAST NAME, STUDENT FIRST NAME, STUDENT SSN) IS GHAD, UG, OR PB - IS NOT INCLUDED IN STATISTICS.

User response: SPIN-OFF file does not show the level of study of the student indicated. If SPIN-OFF file is loaded from test data, an error is probably present in the test data. If SPIN-OFF file is loaded with FTU Student Record Data, then consult with the Data Base Administrator.

Note: Statistics reports will not include this student.

03 MSG: * CANNOT DETERMINE LAST QRT ATTENDED FOR (STUDENT LAST NAME, STUDENT FIRST NAME, STUDENT SSN). CHECK SPIN-OFF DATA AND CURRENT QRT INPUT.

User Response: SPIN-OFF file does not show the student indicated as being enrolled in the quarter specified on the 'Y' card or any previous quarter. If SPIN-OFF file is loaded with test data, an error is probably present in the test data. If SPIN-OFF file is loaded with FTU Student Record Data, then check for incorrect data on the 'Y' card. If 'Y'
card is correct consult the Data Base Administrator.  
Note: Statistics reports will not include this student.

04 MSG: * CANNOT DETERMINE MAJOR OF (STUDENT LAST NAME, 
STUDENT FIRST NAME, STUDENT SSN) DATA NOT INCLUDED 
IN STATISTICS.

User Response: SPIN-OFF file indicates that the subject student is not a MATH, COMP, or STAT major. If SPIN-OFF file is loaded with test data, an error is probably present in the test data. If SPIN-OFF file is loaded with FTU Student Record Data, then consult with the Data Base Administrator.  
Note: Statistics Report will not include this student.

05 MSG: CANNOT LOCATE RECORD TO BE CHANGED FOR SSN (SSN FROM 
INPUT CARD) THE FOLLOWING INPUT CARD DISREGARDED: 
(CARD IMAGE).

User Response: A request was made to delete or alter data in a student record in the MATH-DATA-BASE, probably with card format 'Z', 'X', or 'D'. The SSN on the change card did not match any record in the MATH-DATA-BASE, and so SIS could not make the correction. Check for proper code in column 1 of the change card, or for incorrect SSN in columns 2-10.
06 MSG: ENTRY TO BE DELETED NOT FOUND IN STATISTICS HIST FILE THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)

User Response: An 'H' change to delete an entry in the Statistics History file was made, and SIS could not locate the entry to be deleted. Check the date on the input card with a current listing for correctness, check the 'H' card format to ensure that the correct data is in columns 2 through 7 of the input card. If an addition was to be made to the Statistics History file, check for zeros incorrectly filling columns 8 through 32 of the input card.

07 MSG: * EXPECTED BUT DID NOT FIND * TO INDICATE END MARKER FOR SPIN-OFF. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT).

User Response: If SPIN-OFF is loaded with FTU Student Record Data, the first two input cards were not a 'Y' card followed by a card with an '*' in column 1. As a result, the SPIN-OFF data base has been damaged. Correct the job deck and rerun SIS after SPIN-OFF has been reloaded. If SPIN-OFF is being loaded with test data, insert an '*' after the test data and rerun SIS.

08 MSG: * EXPECTED TO FIND * AT END OF MATH CHANGE CARD INPUT FOUND INSTEAD THE FOLLOWING DATA WHICH IS DISRE-
GARDED: * CARD IMAGE: (CARD TEXT).

User Response: A card with the asterisk in column 1 must immediately follow the last MATH-DATA-BASE change card. SIS found instead a character which was not a change request nor an asterisk. If a card with an asterisk in column 1 is missing, place one in the DECK (figure 13). If the card text indicates that it is for a MATH-DATA-BASE change, check the card format for correct code in column 1, and resubmit change. If the disregarded card was a report request, the report will not be generated.

09 MSG: ILLEGAL ENTRY ON HISTORY ALTERATION REQUEST. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT).
User Response: An 'H' card contains illegal data. Check proper format for 'H' card, repunch card and rerun SIS.

10 MSG: ILLEGAL ENTRY ON REPORT REQUEST. THE FOLLOWING INPUT CARD DISREGARDSED: (CARD TEXT)
User Response: Column 1 of the input card contained a valid character to generate a report, however other extraneous data was found on the card. Check the input card for extraneous data (if it is a report request) or for an incorrect code in column 1 (if it is not a report request).
11 MSG: ILLEGAL ENTRY ON YEAR AND QUARTER INPUT CARD THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)
User Response: 'Y' card contains illegal data in columns 5 through 80 and was out of place. Probable cause is wrong use of 'Y' in column 1. Check card and correct data in column 1.

12 MSG: ILLEGAL INPUT ON ADVISOR INPUT CARD THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)
User Response: 'V' card has blanks or alphabetic characters in columns 2 through 10 or numeric data appears in columns 11 through 80. Check 'V' format and correct the card (if input was intended to be a 'V' card) or check for an incorrect code in column 1.

13 MSG: INVALID CODE IN COLUMN 1 OF INPUT CARD. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)
User Response: An unauthorized code was entered in column 1 of input card. Check proper format and enter correct code on card.

14 MSG: * INVALID CURRENT QUARTER SPECIFIED ON INPUT CARD - PROCESSING TERMINATING. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)
User Response: 'Y' card contained an invalid quarter code. Check quarter code list (appendix B) for
correct conversion data, make correction and rerun SIS.

**MSG:** INVALID SSN ON INPUT CARD. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)

User Response: Check input card SSN for blanks or non-numeric data in columns 2 through 10.

**MSG:** MEMBERSHIP ENTRY TO BE DELETED NOT FOUND. THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)

User Responses: A request was made to delete an association membership entry, however the entry could not be found. Check for correct SSN in columns 2 through 10 of input card, then ensure that the data to be deleted is exactly the same on both the input card and a current listing. Check to make sure that the correct code was entered in column 1 of the input card.

**MSC:** **MISSING OR OUT OF SEQUENCE CURRENT QUARTER CARD PROCESSING TERMINATING THE FOLLOWING INPUT CARD DISREGARDED: (CARD TEXT)

User Response: 'Y' card was not found as the first card of the job deck as required. SIS has been aborted. Check job deck for proper placement of 'Y' card, then rerun SIS.

**MSG:** **NO INPUT FROM STUD DATA BASE.

User Response: While attempting to generate a
report, SIS found no data in the STUD-DATA-BASE. Probable cause is no data in SPIN-OFF file. If SPIN-OFF is loaded with test data, ensure that input deck is correctly formatted, then rerun SIS. If SPIN-OFF is loaded with FTU Student Record Data, then ensure that the first two cards of the job deck contain a 'Y' card immediately followed by a card with an '*' in column 1. If not, SPIN-OFF file has been damaged, and it must be reloaded before SIS is rerun. If the job deck is correct then attempt to rerun SIS. If the same error message is generated, consult with the Data Base Administrator.

19 MSG: * SSN (SSN OF STUDENT) NOT FOUND IN FTU SUPPLIED DATA. VERIFY SSN.
User Response: A student listed in the MATH-DATA-BASE (by SSN) cannot be located in SPIN-OFF file. If SPIN-OFF is loaded with test data, incorrect test data is the probable cause. If SPIN-OFF is loaded from FTU Student Record Data, then verify the correctness of the student's SSN in the MATH-DATA-BASE. If the student is a COMP, MATH, or STAT major, then consult with the Data Base Administrator. Note: Statistics report will not include this student.
<table>
<thead>
<tr>
<th>Academic Quarter</th>
<th>Current Quarter Code Recognized by SIS</th>
<th>Sequential Quarter</th>
</tr>
</thead>
<tbody>
<tr>
<td>F68</td>
<td>691</td>
<td>01</td>
</tr>
<tr>
<td>W69</td>
<td>692</td>
<td>02</td>
</tr>
<tr>
<td>S69</td>
<td>693</td>
<td>03</td>
</tr>
<tr>
<td>U69</td>
<td>694</td>
<td>04</td>
</tr>
<tr>
<td>F69</td>
<td>701</td>
<td>05</td>
</tr>
<tr>
<td>W70</td>
<td>702</td>
<td>06</td>
</tr>
<tr>
<td>S70</td>
<td>703</td>
<td>07</td>
</tr>
<tr>
<td>U70</td>
<td>704</td>
<td>08</td>
</tr>
<tr>
<td>F70</td>
<td>711</td>
<td>09</td>
</tr>
<tr>
<td>W71</td>
<td>712</td>
<td>10</td>
</tr>
<tr>
<td>S71</td>
<td>713</td>
<td>11</td>
</tr>
<tr>
<td>U71</td>
<td>714</td>
<td>12</td>
</tr>
<tr>
<td>F71</td>
<td>721</td>
<td>13</td>
</tr>
<tr>
<td>W72</td>
<td>722</td>
<td>14</td>
</tr>
<tr>
<td>S72</td>
<td>723</td>
<td>15</td>
</tr>
<tr>
<td>U72</td>
<td>724</td>
<td>16</td>
</tr>
<tr>
<td>F72</td>
<td>731</td>
<td>17</td>
</tr>
<tr>
<td>W73</td>
<td>732</td>
<td>18</td>
</tr>
<tr>
<td>S73</td>
<td>733</td>
<td>19</td>
</tr>
<tr>
<td>U73</td>
<td>734</td>
<td>20</td>
</tr>
<tr>
<td>F73</td>
<td>741</td>
<td>21</td>
</tr>
<tr>
<td>W74</td>
<td>742</td>
<td>22</td>
</tr>
<tr>
<td>S74</td>
<td>743</td>
<td>23</td>
</tr>
<tr>
<td>U74</td>
<td>744</td>
<td>24</td>
</tr>
<tr>
<td>Academic Quarter</td>
<td>Current Quarter Code Recognized by SIG</td>
<td>Sequential Quarter</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>P74</td>
<td>751</td>
<td>25</td>
</tr>
<tr>
<td>W75</td>
<td>752</td>
<td>26</td>
</tr>
<tr>
<td>S75</td>
<td>753</td>
<td>27</td>
</tr>
<tr>
<td>U75</td>
<td>754</td>
<td>28</td>
</tr>
<tr>
<td>F75</td>
<td>761</td>
<td>29</td>
</tr>
<tr>
<td>W76</td>
<td>762</td>
<td>30</td>
</tr>
<tr>
<td>S76</td>
<td>763</td>
<td>31</td>
</tr>
<tr>
<td>U76</td>
<td>764</td>
<td>32</td>
</tr>
<tr>
<td>F76</td>
<td>771</td>
<td>33</td>
</tr>
<tr>
<td>W77</td>
<td>772</td>
<td>34</td>
</tr>
<tr>
<td>S77</td>
<td>773</td>
<td>35</td>
</tr>
<tr>
<td>U77</td>
<td>774</td>
<td>36</td>
</tr>
<tr>
<td>F77</td>
<td>781</td>
<td>37</td>
</tr>
<tr>
<td>W78</td>
<td>782</td>
<td>38</td>
</tr>
<tr>
<td>S78</td>
<td>783</td>
<td>39</td>
</tr>
<tr>
<td>U78</td>
<td>784</td>
<td>40</td>
</tr>
</tbody>
</table>
Appendix C

SIS LISTING

(LISTING IS EXTERNAL TO THE BOUND REPORT)
Appendix D

HIPO INPUT-PROCESS-OUTPUT (I-P-O) DIAGRAMS
1. Provide administrative and counselling data about students.

Other subsystems to be added.

Output:

- STUDENT DATA LIST
- FACULTY ADVISOR LISTING
- ADVISEE LIST
- STUDENT STATISTICS
1. Load current QTR, current date, Decode and verify MATH DEPT INPUT DATA then, based on results, perform 2, 3, or 4.

2. Maintain Math Dept data bases. 2.1

3. Capture FTU student data. 2.2

4. Generate student reports. 2.3

RETURN
2.0

2.1. If INPUT DATA is for changing MATH-DEPT-STUD-DB:
   A. Read changes into MATH-CHANGE file and sort.
   B. Search MATH DEPT DATA BASE on key SSN for student record.
   C. If student record not found create new record.
   D. If student record found change record.
   E. Rewrite MATH-REORG into MATH DEPT DATA BASE.
   2.1.3 Update STAT-HIST-DB if requested.
   RETURN

A

MATH-CHANGE

MATH-REORG

MATH DEPT DATA BASE

STAT HIST

DATE-BUFF

ACCUMULATORS
2.1

1. Null all fields in new record.

2. Move input data to appropriate fields of new record.

3. Write new record in MATH REORG.
1. If data is to be added to student record move input data to appropriate fields in student record.
2. If data is to be deleted, delete data in student record.
3. Rewrite student record.
1. If CODE = 'H' then move input data to appropriate accumulators, and date to DATE-BUFF.

2. Search STAT HIST file for record with same date as DATE-BUFF.
   A. If found rewrite STAT HIST record.
   B. If not found write new STAT-HIST record.

3. Search STAT HIST file for records with zeros in 'COMP/UG' field, delete them, move other records to HIST-REORG.

4. Sort HIST-REORG file in ascending order on key 'DATE' into STAT-HIST.
1. Perform 2 if FTU data is available or 3 if it must be manually input.

2. Capture FTU data on STUD-DATA-SPIN-OFF.

3. Manually load FTU data on STUD-DATA-SPIN-OFF.

4. Read STUD-DATA-SPIN-OFF file and create and enter data, in correct format, in STUD DATA BASE.
Note: SIS will accept data from SPIN-OFF if loaded by the DBA, without SIS modification.
1. Read a card.

2. Compare SSN of Card Data with current record on SPIN-OFF.
   A. If record not same write new record on STUD-DATA-SPIN-OFF, nulling all fields then entering data into correct fields.
   B. If record same add data to record in appropriate fields.

3. If last card

1. Advance to first record in SPIN-OFF.
   DO:
2. Read SPIN-OFF record.
3. Move corresponding SPIN-OFF-REC to STUD-DB-REC.
4. Determine most recent quarter student attended PTU and if graduated:
   A. Move INDEX of SPIN-OFF-REC to INDEX-ARRAY.
   B. Initialize CTR to 0.
   DO:
   C. If INDEX-ARRAY(indexed by CURRENT-QTR-CODE - CTR) ≠ '*' then
   D. CTR = CTR + 1.
   E. If GRD-IND of INDEX of SPIN-OFF-REC(indexed by CURRENT-QTR-CODE - CTR) = '1' or '2' student has graduated, enter 'Y' in GRAD of STUD-DB-REC else enter 'N'.
   F. If CTR = 0 student is currently enrolled, CURRENT-QTR to PTU-LAST-QTR of 3STUD-DB-REC.
   G. If CTR ≠ 0 student is not currently enrolled, set PTU-LAST-QTR of STUD-DB-REC = QTR-TAB(indexed by CURRENT-QTR-CODE - CTR).
1. Merge STUD DATA BASE (with PTU data) and MATH MATH DEPT DATA BASE giving 3TUD DATA BASE.

2. Identify type of report required then (using utility sort where required):
   A. If student statistics report is requested
   B. If STUDENT DATA ROSTER or FACULTY ADVISOR LISTING is requested then sort STUD DATA BASE on keys MAJOR, GRAD-UG-PS, NAME.
   C. If ADVISOR LIST is requested then sort STUD DATA BASE on keys ADVISOR-SSN, NAME.

3. Compute and print STUDENT STATISTICS if requested.

4. Print student reports.

5. Return.
### PROCESS

1. Advance to the first record in the MATH DEPT DATA BASE and read.

2. Advance to the first record in the STUD DATA BASE and read.

**DO:**

3. If SSN of STUD DATA BASE record is the same as SSN of the MATH DEPT DATA BASE record then:
   
   A. Copy information in MATH DEPT DATA BASE record onto STUD DATA BASE record in proper format.
   
   B. Read next record in MATH DEPT DATA BASE.

4. If SSN of STUD-REC-M < SSN of STUD-DB-REC print ERR-MSG, read next STUD-REC-M.

5. If end of MATH DEPT DATA BASE or end of STUD DATA BASE

6. Return
1. Initialize accumulators to zero and DATE-BUFF.
2. Advance paper to top of page.
3. Print headings for STUDENT STATISTICS report.
4. Advance to first record of STUD DATA BASE.
   DO:
5. Read record from STUD DATA BASE.
6. Check record's MAJOR and LEVEL fields and increment appropriate accumulator.
7. If last record:
8. Advance to next record.

Input

Process

Output

HEADINGS TAB
STUD STAT
STUD DATA BASE

STUDENT STATISTICS
STUD DATA BASE RECORD
ACCUMULATORS
COMP/UG
COMP/GR
COMP/PB
MATH/UG
MATH/GR
MATH/PB
STAT/UG

STAT HIST
2.1.3

DATE BUFF
10. Advance to first record in STAT HIST file.
   DO:
   11. Read STAT HIST file record into appropriate accumulators.
   12. Move record data to output buff.
   13. Set COMPTOT = COMP/UG + COMP/GR + COMP/PB.
   14. Set MATHTOT = MATH/UG + MATH/GR + MATH/PB.
   15. Set STATTOT = STAT/UG.
   16. Set TOTSTUDS = COMPTOT + MATHTOT + STATTOT.
   17. Move output formats and accumulator data to appropriate output buffers.
   18. Print student statistics report from output buffer.
   19. If last record
   20. Locate next record in STAT HIST file.
1. Determine type of report requested. Then, for currently enrolled students:
   A. If STUDENT DATA ROSTER is requested, set CODE=R.

   PRINT-RTN

   4.1

   B. If FACULTY ADVISOR LISTING is requested, set CODE=F.

   PRINT-RTN

   4.1

   C. If ADVISEE LIST is requested, set CODE=A.

   PRINT-RTN

   4.1

2. If STUDENT DATA ROSTER is requested then:
   A. Sort STUD DATA BASE on keys ADVISOR, NAME.
   B. Set CODE=S.
   C. Then for currently enrolled students.

   PRINT-RTN

   4.1

RETURN

REPORTS
Input

KODE

HEADINGS

OUTPUT FORMAT

STUD DATA BASE

Process

1. Read first record from STUD DATA BASE into STUD-REC.
2. Load break field of STUD-DB-REC into BREAK 1,2,3.
3. Advance to top of page.
4. Print HEADINGS.

DO:

5. Skip two lines.
6. Place proper fields of STUD-REC into OUTBUFS.
7. Print OUTBUF.
8. If last record in STUD DATA BASE
9. Read next STUD-DB-REC.
10. If breakfield is not the same as BREAK we have reached break.
11. Return.

Output

STUD-DB-REC

BREAK 1,2,3.

REPORT

OUTBUFS

RETURN

Diagram ID: 4.1
Name: PRINT-RTN
Description: PRINT SUBROUTINE

Date: 3/6/76
Page: 1 of 1

Printed in U.S.A.
LISTING OF DECK # 1 (SIS RUN DECK)
//COMP69RP JOB (1770,3376,NCLZ,FTU,5,5),'ROGER SIFRIT',CLASS=8. 
// MSGLEVEL=(1,1)
// JOBPARM PSS=NO,F=1410,K=61
// EXEC COBOLCG 
IDENTIFICATION DIVISION.
PROGRAM-ID. CREATE-MATH-DEPT-STUD-DB.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT MATH-DEPT-STUD-DB ASSIGN TO DA-3330-S-DISKMDR.
DATA DIVISION.
FILE SECTION.
FD MATH-DEPT-STUD-DB LABEL RECORD IS STANDARD RECORDING MODE F
DATA RECORD IS STUD-REC-M.
01 STUD-REC-M.
  02 ATTEND-DATA PIC X(100).
  02 MBR-DATA PIC X(70).
  02 SSN PIC 9(09).
  02 ADVISOR-NAME PIC X(18).
  02 ADVISOR-SSN PIC 9(09).
PROCEDURE DIVISION.
OPEN OUTPUT MATH-DEPT-STUD-DB.
MOVE SPACES TO STUD-REC-M.
MOVE ZEROS TO SSN OF STUD-REC-M.
WRITE STUD-REC-M.
CLOSE MATH-DEPT-STUD-DB.
STOP RUN.
//GO,DISKMDR DD DSN=FTU,OU,P1770,MD8,
// SPACE=(TRK,(1,1)),
// DISP=(NEW,KEEP,CATLG),
// UNIT=DISK*VOL=SER=FTUPK1

LISTING OF DECK # 3

(PFOR CREATION OF MATH-DEPT-STUD-DB FILE)
LISTING OF DECK # 4
(FOR CREATION OF STAT-HIST-DB FILE)
IDENTIFICATION DIVISION.

PROGRAM-ID. STUDENT INFORMATION SYSTEM.

AUTHOR. ROGER W SIFRIT.

INSTALLATION. FTU.

DATE-WRITTEN. MARCH 31, 1976.

DATE-Compiled. JUN 3, 1976.

SECURITY. THE REPORTS PRODUCED BY THIS PROGRAM CONTAIN INFORMATION WHICH IS PERSONAL IN NATURE.

REMARKS.
0012 ENVIRONMENT DIVISION.
0013 CONFIGURATION SECTION.
0014 SOURCE-COMPUTER, IBM-360-65.
0015 OBJECT-COMPUTER, IBM-360-65.
0016 INPUT-OUTPUT SECTION.
0017 FILE-CONTROL.
0018 SELECT CARD-IN ASSIGN TO UT-S-READER.
0019 SELECT PRINT-FILE ASSIGN TO UT-S-PROUT.
0020 SELECT MATH-CHANGE ASSIGN TO UT-S-DISKC.
0021 SELECT SPIN-OFF ASSIGN TO DA-3330-S-DISKSP.
0022 SELECT STUD-DATA-RAF ASSIGN TO DA-3330-S-DISKSD.
0023 SELECT STAT-HIST-DR ASSIGN TO DA-3330-S-HISTDR.
0024 SELECT MATH-DEPT-STUD-DR ASSIGN TO DA-3330-S-DISKMD.
0025 SELECT SORT-WORK-DR ASSIGN TO DA-3330-S-SORTWK01.
0026 SELECT SORT-WORK-HDR ASSIGN TO DA-3330-S-SORTWK01.
0027 SELECT INPUT-SORT-FILE ASSIGN TO DA-3330-S-SORTWK01.
0028 SELECT MATH-REORG ASSIGN TO DA-3330-S-DISKMR.
0029 SELECT HIST-REORG ASSIGN TO DA-3330-S-DISKHR.
0030 INPUT-OUTPUT SECTION.
0031 SAME SORT AREA FOR SORT-WORK-SDR SORT-WORK-HDR SORT-WORK-SPO
0032 INPUT-SORT-FILE.
**DATA DIVISION.**

**FILE SECTION.**

<table>
<thead>
<tr>
<th>FD</th>
<th>CARD-IN LABEL RECORD IS OMITTED RECORDING MODE IS F RECORD</th>
</tr>
</thead>
</table>

<p>| 00040      | 01 CODE-CHECK. |
| 00041      | 02 COOD PIC X(01). |
| 00042      | 02 REST PIC X(79). |
| 00043      | 01 FTU-ATTEND-Q. |
| 00044      | 02 FILLER PIC X(01). |
| 00045      | 02 SSN PIC 9(09). |
| 00046      | 02 ATTEND-DATA PIC X(70). |
| 00047      | 02 FILLER PIC X(01). |
| 00048      | 02 FILLER PIC X(68). |
| 00049      | 02 MRR-DELETE-X. |
| 00050      | 02 ASSN-MBR-M. |
| 00051      | 02 FILLER PIC X(01). |
| 00052      | 02 SSN PIC 9(09). |
| 00053      | 02 MRR-DELETE-X. |
| 00054      | 02 FILLER PIC X(01). |
| 00055      | 02 MRR-DELETE-X. |
| 00056      | 02 INDEX-4. |
| 00057      | 02 INDEX-4. |
| 00058      | 02 INDEX-4. |
| 00059      | 02 INDEX-4. |
| 00060      | 02 INDEX-4. |
| 00061      | 02 INDEX-4. |
| 00062      | 02 INDEX-4. |
| 00063      | 02 INDEX-4. |
| 00064      | 02 INDEX-4. |
| 00065      | 02 INDEX-4. |
| 00066      | 02 INDEX-4. |
| 00067      | 02 INDEX-4. |
| 00068      | 02 INDEX-4. |
| 00069      | 02 INDEX-4. |
| 00070      | 02 INDEX-4. |
| 00071      | 02 INDEX-4. |
| 00072      | 02 INDEX-4. |
| 00073      | 02 INDEX-4. |
| 00074      | 02 INDEX-4. |
| 00075      | 02 INDEX-4. |
| 00076      | 02 INDEX-4. |
| 00077      | 02 INDEX-4. |
| 00078      | 02 INDEX-4. |
| 00079      | 02 INDEX-4. |
| 00080      | 02 INDEX-4. |
| 00081      | 02 INDEX-4. |
| 00082      | 02 INDEX-4. |
| 00083      | 02 INDEX-4. |
| 00084      | 02 INDEX-4. |
| 00085      | 02 INDEX-4. |
| 00086      | 02 INDEX-4. |
| 00087      | 02 INDEX-4. |
| 00088      | 02 INDEX-4. |
| 00089      | 02 INDEX-4. |</p>
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00090</td>
<td>PARENTS-ADDRESS-3.</td>
</tr>
<tr>
<td>00091</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00092</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00093</td>
<td>02 HOME-STREET</td>
</tr>
<tr>
<td>00094</td>
<td>02 HOME-CITY</td>
</tr>
<tr>
<td>00095</td>
<td>02 HOME-STATE</td>
</tr>
<tr>
<td>00096</td>
<td>02 HOME-ZIP</td>
</tr>
<tr>
<td>00097</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00098</td>
<td>STUD-ADDRESS-2.</td>
</tr>
<tr>
<td>00099</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00100</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00101</td>
<td>02 HOME-STREET</td>
</tr>
<tr>
<td>00102</td>
<td>02 HOME-CITY</td>
</tr>
<tr>
<td>00103</td>
<td>02 HOME-STATE</td>
</tr>
<tr>
<td>00104</td>
<td>02 HOME-ZIP</td>
</tr>
<tr>
<td>00105</td>
<td>02 HOME-PHONE</td>
</tr>
<tr>
<td>00106</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00107</td>
<td>FTU-ATTEND-DELETE-Z.</td>
</tr>
<tr>
<td>00108</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00109</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00110</td>
<td>02 ATTEND-DATA</td>
</tr>
<tr>
<td>00111</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00112</td>
<td>YEAR-AND-QUARTER-Y.</td>
</tr>
<tr>
<td>00113</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00114</td>
<td>02 DATA-Y.</td>
</tr>
<tr>
<td>00115</td>
<td>05 SCHOOL-YEAR</td>
</tr>
<tr>
<td>00116</td>
<td>05 SCHOOL-QUARTER</td>
</tr>
<tr>
<td>00117</td>
<td>02 FILLER-Y</td>
</tr>
<tr>
<td>00118</td>
<td>ADVISOR-V.</td>
</tr>
<tr>
<td>00119</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00120</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00121</td>
<td>02 ADVISOR-NAME</td>
</tr>
<tr>
<td>00122</td>
<td>02 ADVISOR-SSN</td>
</tr>
<tr>
<td>00123</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00125</td>
<td>FD MATH-CHANGE LABEL RECORD IS STANDARD RECORDING MODE F</td>
</tr>
<tr>
<td>00126</td>
<td>DATA RECORDS ARE CODE-CHECK-W, FTU-ATTEND-W, ASSN-MBR-W,</td>
</tr>
<tr>
<td>00127</td>
<td>MBR-DELETE-W, FTU-ATTEND-DELETE-W, ADVISOR-W,</td>
</tr>
<tr>
<td>00128</td>
<td>MATH-CHANGE-REC.</td>
</tr>
<tr>
<td>00129</td>
<td>01 CODE-CHECK-W.</td>
</tr>
<tr>
<td>00130</td>
<td>02 CWD</td>
</tr>
<tr>
<td>00131</td>
<td>02 REST-W</td>
</tr>
<tr>
<td>00132</td>
<td>FTU-ATTEND-W.</td>
</tr>
<tr>
<td>00133</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00134</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00135</td>
<td>02 ATTEND-DATA</td>
</tr>
<tr>
<td>00136</td>
<td>02 MBR-DATA</td>
</tr>
<tr>
<td>00137</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00138</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00139</td>
<td>02 MBR-DATA</td>
</tr>
<tr>
<td>00140</td>
<td>MBR-DELETE-W.</td>
</tr>
<tr>
<td>00141</td>
<td>02 FILLER</td>
</tr>
<tr>
<td>00142</td>
<td>02 SSN</td>
</tr>
<tr>
<td>00143</td>
<td>02 ASSN-NAME</td>
</tr>
<tr>
<td>00144</td>
<td>FTU-ATTEND-DELETE-W.</td>
</tr>
<tr>
<td>00145</td>
<td>02 FILLER</td>
</tr>
</tbody>
</table>
00146 02 SSN PIC 9(09).
00147 02 ATTEND-DATA PIC X(04).
00148 02 FILLER PIC X(66).
00149 02 ADVISOR=W.
00150 02 FILLER PIC X(01).
00151 02 SSN PIC 9(09).
00152 02 ADVISOR-NAME PIC X(18).
00153 02 ADVISOR-SSN PIC 9(09).
00154 02 FILLER PIC X(43).
00155 01 MATH-CHANGE-REC.
00156 05 FILLER PIC X(01).
00157 05 SSN PIC 9(09).
00158 05 FILLER PIC X(70).

FD STUD-DATA-BASE LABEL RECORD IS STANDARD RECORDING MODE IS F

FD STUD-DB-REC.

FD PRINT-FILE LABEL RECORDS OMITTED DATA RECORD IS PRINT-LINE.

FD SPIN-OFF LABEL RECORD IS STANDARD RECORDING MODE IS F

FD SPIN-OFF-REC.
00200 05 FTU-GPA  PIC 9V999.
00201 02 SEX  PIC X(01).
00202 02 INDEX  PIC X(02).
00203 05 GRD-IND  PIC X(01).
00204 05 START-SUR  PIC X(02).
00205 05 END-SUR  PIC X(02).
00206 02 HOME-STREET  PIC X(20).
00207 02 HOME-CITY  PIC X(20).
00208 02 HOME-STATE  PIC X(02).
00209 02 HOME-ZIP  PIC X(05).
00210 02 HOME-PHONE  PIC X(10).
00211 02 KIN-STREET  PIC X(20).
00212 02 KIN-CITY  PIC X(20).
00213 02 KIN-STATE  PIC X(02).
00214 02 KIN-ZIP  PIC X(05).
00215 02 MARITAL-STATUS  PIC X(01).
00216 01 STAT-REC.
00217 03 QTR  PIC 9(03).
00218 03 COMP-UG  PIC 9(04).
00219 03 COMP-GR  PIC 9(04).
00220 03 COMP-PB  PIC 9(04).
00221 03 MATH-UG  PIC 9(04).
00222 03 MATH-GR  PIC 9(04).
00223 03 MATH-PB  PIC 9(04).
00224 03 STAT-UG  PIC 9(04).
00225 03 STAT-UG-PH  PIC 9(04).
00226 02 LAST-NAME  PIC X(30).
00227 02 FPST-MOLE-NAM  PIC X(30).
00228 02 SSN  PIC 9(09).
00229 02 GRAD-UG-PH  PIC X(01).
00230 02 MAJOR  PIC X(04).
00231 02 FTU-GPA  PIC 9V999.
00232 02 SEX  PIC X(01).
00233 02 GRAD  PIC X(01).
00234 02 FTU-LAST-QTR  PIC 9(03).
00235 02 HOME-STREET  PIC X(20).
00236 02 HOME-CITY  PIC X(20).
00237 02 HOME-STATE  PIC X(02).
00238 02 HOME-ZIP  PIC X(05).
00239 02 HOME-PHONE  PIC X(10).
00240 02 KIN-STREET  PIC X(20).
00241 02 KIN-CITY  PIC X(20).
00242 02 KIN-STATE  PIC X(02).
00243 02 KIN-ZIP  PIC X(05).
00244 02 MATH-IND  PIC X(01).
00245 02 ATTEND-DATE  PIC X(70).
00246 02 MPR-DATA  PIC X(70).
00247 02 ADVISOR-NAME  PIC X(18).
00248 02 ADVISOR-SSN  PIC 9(09).
00307      02 SEX       PIC X(01).
00308      02 INDX      PIC X(200).
00309      05 GRO-IND PIC X(01).
00310      05 START-SUR PIC X(02).
00311      05 END-SUR PIC X(02).
00312      02 HOME-STREET PIC X(20).
00313      02 HOME-CITY PIC X(20).
00314      02 HOME-STATE PIC X(02).
00315      02 HOME-ZIP PIC 9(05).
00316      02 HOME-PHONE PIC 9(10).
00317      02 KIN-STREET PIC X(20).
00318      02 KIN-CITY PIC X(20).
00319      02 KIN-STATE PIC X(02).
00320      02 KIN-ZIP PIC 9(05).
00321      02 MARRITAL-STATUS PIC X(01).

FD MATH-REORG LABEL RECORD IS STANDARD DATA RECORD IS

00323      01 MATH-REORG-REC.
00324      02 ATTEND-DATA PIC X(100).
00325      02 MAR-DATA PIC X(70).
00326      02 SSN PIC 9(09).
00327      02 ADVISOR-NAME PIC X(18).
00328      02 ADVISOR-SSN PIC 9(09).

00334      WORKING-STORAGE SECTION.

00335      77 ERR-MSG-LINE PIC X(133) VALUE SPACES.
00336      77 MF4-AREA PIC X(133).
00337      77 CURRENT-QTR-CODE PIC 9(02).
00338      77 CURRENT-QTR PIC 9(03).
00339      77 QTR PIC 9(10).
00340      77 PTR PIC 9(10).
00341      77 MATH-CHANGE-MADE PIC X(01) VALUE 'Y'.
00342      77 SUBSCRIPT PIC 9(10) VALUE IS ZERO, USAGE COMP.
00343      77 HEADINGS PIC X(20).
00344      77 OUTPUT-FORMAT PIC X(20).
00345      77 BREAK PIC X(20).
00346      77 DATE-RUFF PIC X(06).
00347      77 YYMMDD PIC X(20).
00348      77 TEMP PIC X(20).
00349      77 ERROR-WARNING PIC X(18) VALUE '*** ERROR WARNING!'.'.
00350      77 BREAK-STATUS PIC X(03) VALUE 'NO'.
00351      77 KONF PIC X(01).
00352      77 BREAK-1 PIC X(04).
00353      77 BREAK-2 PIC X(01).
00354      77 BREAK-3 PIC 9(09).
00355      77 VALIDITY-INDICATOR PIC X(07).
00356      77 SSNUM PIC 9(09).
00357      77 QTR-RUFF PIC 9(03) VALUE ZEROS.
00358      77 DELETE-CHAR PIC X(01) VALUE SPACES.
00359      01 RECORD-RUFF.
00360      02 ATTEND-DATA PIC X(100).
00362 02 MRR-DATA PIC X(70).
00363 02 SSN PIC 9(09).
00364 02 ADVISOR-NAME PIC X(18).
00365 02 ADVISOR-SSN PIC 9(09).
00366
00367 01 FTU-LAST-QTR-EDIT.
00368 05 LAST-YR-ATT-EDIT PIC 9(02).
00369 05 FILLER PIC X(01).
00370 01 ACCUMULATORS.
00371 03 COMP-UG PIC 9(04) VALUE ZERO.
00372 03 COMP-GR PIC 9(04) VALUE ZERO.
00373 03 COMP-PB PIC 9(04) VALUE ZERO.
00374 03 COMPTOT PIC 9(04) VALUE ZERO.
00375 03 MATH-GR PIC 9(04) VALUE ZERO.
00376 03 MATH-PB PIC 9(04) VALUE ZERO.
00377 03 MATHTOT PIC 9(04) VALUE ZERO.
00378 03 STAT-JUG PIC 9(04) VALUE ZERO.
00379 03 STATTOT PIC 9(04) VALUE ZERO.
00380 03 FILLER PIC X(01).
00381 03 TOTSTDNS PIC 9(04) VALUE ZERO.
00382 01 FORMER-STUD-HEAD.
00383 05 FILLER PIC X(39) VALUE SPACES.
00384 05 FILLER PIC X(59) VALUE 'NOT CURRENTLY ENROLLED'.
00385 05 FILLER PIC X(38) VALUE SPACES.
00386 01 INDEX-CONVERT.
00387 05 INDEX-DATA PIC X(70).
00388 05 FILLER PIC X(130) VALUE SPACES.
00389
00390 01 ATTEND-ARRAY PIC X(120).
00391 01 ATTEND-TAR REDEFINES ATTEND-ARRAY.
00392 02 QTR-ATTEND PIC X(04) OCCURS 10 INDEXED BY Q.
00393 01 MRR-TAR-FIELD PIC X(100).
00394 01 MRR-TAR-REDEFINES MRR-TAR-FIELD.
00395 02 MRR PIC X(10) OCCURS 10 INDEXED BY M.
00396 01 RECORD-CHANGE.
00397 02 FILLER PIC X(14) VALUE 'STUDENT SSN: ','.
00398 02 SSN PIC 9(09).
00399 02 FILLER PIC X(16) VALUE 'FIELD CHANGED: ','.
00400 02 FILLER PIC X(12).
00401 02 NAME-DATA PIC X(92).
00402 01 STUD-REC-TEMP.
00403 02 ATTEND-DATA PIC X(40) VALUE SPACES.
00404 02 MRR-DATA PIC X(70) VALUE SPACES.
00405 02 SSN PIC 9(09) VALUE ZEROS.
00406 02 ADVISOR-NAME PIC X(18) VALUE SPACES.
00407 02 ADVISOR-SSN PIC 9(09) VALUE ZEROS.
00408 01 TODAYS-DATE.
00409 01 TQDAYS-DATE.
00410 02 CURRENT-DATY.
00411 02 YY PIC 9(02).
00412 02 MM PIC 9(02).
00413 02 DD PIC 9(02).
00414 01 MONT-TAR-REDEFINES MONT-TAR-DATA.
00415 01 MONT-TAR-REDEFINES MONT-TAR-DATA.
00416 01 MONTH-ALPHA PIC X(03) OCCURS 12 TIMES.
00417 02 MONT-ALPHA PIC X(03) OCCURS 12 TIMES.
00418 01 DATE-PRINT-TRM.
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00419</td>
<td>02 MONTH PIC X(03).</td>
</tr>
<tr>
<td>00420</td>
<td>02 FILLER PIC X(01) VALUE SPACES.</td>
</tr>
<tr>
<td>00421</td>
<td>02 DD PIC 9(02).</td>
</tr>
<tr>
<td>00422</td>
<td>02 FILLER PIC X(03) VALUE '19'.</td>
</tr>
<tr>
<td>00423</td>
<td>02 YY PIC 9(02).</td>
</tr>
<tr>
<td>00424</td>
<td>01 QTR-CONVERT-TAR.</td>
</tr>
<tr>
<td>00425</td>
<td>05 FILLER PIC X(78) VALUE '69169269369470170270'.</td>
</tr>
<tr>
<td>00426</td>
<td>05 FILLER PIC 9(02).</td>
</tr>
<tr>
<td>00427</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00428</td>
<td>05 FILLER PIC X(60) VALUE '1'.</td>
</tr>
<tr>
<td>00429</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00430</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00431</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00432</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00433</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00434</td>
<td>05 FILLER PIC X(60) VALUE '17'.</td>
</tr>
<tr>
<td>00435</td>
<td>01 QTR-TAR REDEFINES QTR-CONVERT-TAR.</td>
</tr>
<tr>
<td>00436</td>
<td>02 YY-QTR OCCURS 88 TIMES INDEXED BY YY.</td>
</tr>
<tr>
<td>00437</td>
<td>05 QTR PIC 9(03).</td>
</tr>
<tr>
<td>00438</td>
<td>01 INDEX-ARRAY-LOAD-FIELD PIC X(201).</td>
</tr>
<tr>
<td>00439</td>
<td>01 INDEX-ARRAY REDEFINES INDEX-ARRAY-LOAD-FIELD.</td>
</tr>
<tr>
<td>00440</td>
<td>02 INDEX-ENTRY OCCURS 40 TIMES.</td>
</tr>
<tr>
<td>00441</td>
<td>05 GBD-IND PIC 9(01).</td>
</tr>
<tr>
<td>00442</td>
<td>05 START-SUR PIC 9(02).</td>
</tr>
<tr>
<td>00443</td>
<td>05 END-SUR PIC 9(02).</td>
</tr>
<tr>
<td>00444</td>
<td>01 PAGE-HEAD-DEPT.</td>
</tr>
<tr>
<td>00445</td>
<td>05 FILLER PIC X(49) VALUE SPACES.</td>
</tr>
<tr>
<td>00446</td>
<td>05 FILLER PIC X(35) VALUE 'DEPARTMENT OF MATHEM'.</td>
</tr>
<tr>
<td>00447</td>
<td>05 FILLER PIC X(49) VALUE SPACES.</td>
</tr>
<tr>
<td>00448</td>
<td>05 FILLER PIC X(49) VALUE SPACES.</td>
</tr>
<tr>
<td>00449</td>
<td>01 TITLE-ROST.</td>
</tr>
<tr>
<td>00450</td>
<td>05 FILLER PIC X(44) VALUE SPACES.</td>
</tr>
<tr>
<td>00451</td>
<td>05 TITLE-LINE PIC X(46).</td>
</tr>
<tr>
<td>00452</td>
<td>05 FILLER PIC X(43) VALUE SPACES.</td>
</tr>
<tr>
<td>00453</td>
<td>01 STUD-STAT-HEAD.</td>
</tr>
<tr>
<td>00454</td>
<td>05 FILLER PIC X(58) VALUE SPACES.</td>
</tr>
<tr>
<td>00455</td>
<td>05 FILLER PIC X(58) VALUE SPACES.</td>
</tr>
<tr>
<td>00456</td>
<td>05 FILLER PIC X(58) VALUE SPACES.</td>
</tr>
<tr>
<td>00457</td>
<td>01 DATF-HEAD.</td>
</tr>
<tr>
<td>00458</td>
<td>05 FILLER PIC X(18) VALUE 'STUDENT ENROLLMENT'.</td>
</tr>
<tr>
<td>00459</td>
<td>05 DATE-OUT PIC X(57) VALUE SPACES.</td>
</tr>
<tr>
<td>00460</td>
<td>05 FILLER PIC X(57) VALUE SPACES.</td>
</tr>
<tr>
<td>00461</td>
<td>01 STUD-STAT-COL-HEAD.</td>
</tr>
<tr>
<td>00462</td>
<td>05 FILLER PIC X(61) VALUE SPACES.</td>
</tr>
<tr>
<td>00463</td>
<td>05 FILLER PIC X(61) VALUE SPACES.</td>
</tr>
<tr>
<td>00464</td>
<td>05 FILLER PIC X(61) VALUE SPACES.</td>
</tr>
<tr>
<td>00465</td>
<td>01 STUD-STAT-UNGRAD.</td>
</tr>
<tr>
<td>00466</td>
<td>05 FILLER PIC X(41) VALUE SPACES.</td>
</tr>
<tr>
<td>00467</td>
<td>05 FILLER PIC X(15) VALUE 'UNDERGRADUATE'.</td>
</tr>
<tr>
<td>00468</td>
<td>05 FILLER PIC X(15) VALUE 'UNDERGRADUATE'.</td>
</tr>
<tr>
<td>00469</td>
<td>05 COMP-UG PIC ZZZ9 VALUE ZEROS.</td>
</tr>
<tr>
<td>00470</td>
<td>05 FILLER PIC X(06) VALUE SPACES.</td>
</tr>
<tr>
<td>00471</td>
<td>05 MATH-UG PIC ZZZ9 VALUE ZEROS.</td>
</tr>
<tr>
<td>00472</td>
<td>05 FILLER PIC X(06) VALUE SPACES.</td>
</tr>
<tr>
<td>00473</td>
<td>05 STAT-UG PIC ZZZ9 VALUE ZEROS.</td>
</tr>
<tr>
<td>00474</td>
<td>05 FILLER PIC X(06) VALUE SPACES.</td>
</tr>
<tr>
<td>00475</td>
<td>05 TOT-UG PIC ZZZ9 VALUE ZEROS.</td>
</tr>
<tr>
<td>Field Name</td>
<td>PIC Type</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(10)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(73)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(50)</td>
</tr>
<tr>
<td>01 CURRENT-OTR-PRINT</td>
<td>X(65)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(03)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(65)</td>
</tr>
<tr>
<td>01 NAME-LINE</td>
<td>X(01)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(18)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(30)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(01)</td>
</tr>
<tr>
<td>06 LAST-YR-ATT</td>
<td>X(02)</td>
</tr>
<tr>
<td>06 LAST-QTR-ATT</td>
<td>X(01)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(10)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(18)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(06)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(01)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(12)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(04)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(01)</td>
</tr>
<tr>
<td>01 ADDR-LINE</td>
<td>X(18)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(02)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(01)</td>
</tr>
<tr>
<td>05 HOME-CITY</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 HOME-STATE</td>
<td>X(XXX)</td>
</tr>
<tr>
<td>05 HOME-ZIP</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 HOME-CITY</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 HOME-STATE</td>
<td>X(XXX)</td>
</tr>
<tr>
<td>05 HOME-ZIP</td>
<td>X(XX)</td>
</tr>
<tr>
<td>01 KIN-PARENT-LINE</td>
<td>X(15)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(08)</td>
</tr>
<tr>
<td>06 KIN-ATTN-LINE</td>
<td>X(35)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(63)</td>
</tr>
<tr>
<td>05 FTU-ATTN-LINE</td>
<td>X(35)</td>
</tr>
<tr>
<td>01 FTU-STATUS</td>
<td>X(08)</td>
</tr>
<tr>
<td>06 FTU-OH-ATTN-LINE</td>
<td>X(15)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(08)</td>
</tr>
<tr>
<td>05 KIN-PARENT-LINE</td>
<td>X(15)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(63)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(35)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(08)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(63)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(35)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(08)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(20)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(05)</td>
</tr>
<tr>
<td>05 FILLER</td>
<td>X(63)</td>
</tr>
</tbody>
</table>
00590 05 ATTEND-DATE PIC X(70).
00591 01 WARNING-MSG. PIC X(40) VALUE "* WARNING: INPUT CAR"
00592 05 FILLER PIC X(95) VALUE SPACES.
00593 05 FILLER PIC X(40) VALUE SPACES.
00594 05 FILLER PIC X(95) VALUE SPACES.
00595 01 ADVISEE-DET-LINE. PIC X(10) VALUE SPACES.
00596 05 FILLER PIC X(25).
00597 05 NAME PIC X(98) VALUE SPACES.
00600    **** REFERENCE HIPO 2.0 ****
00601    PROCEDURE DIVISION.
00602    PERFORM ACCEPT-DATF-FROM-SYSTEM.
00603    OPEN INPUT CARD-IN.
00604    READ CARD-IN AT END GO TO EOJ.
00605    PERFORM LOAD-CURRENT-QTR-DATA.
00606    IF KODE = 'I' GO TO EOJ.
00607    READ CARD-IN AT END GO TO EOJ.
00608    PERFORM LOAD-FTU-DATA THRU END-FTU-DATA-SECTION.
00609    READ CARD-IN AT END GO TO EOJ.
00610    PERFORM CARD-FORMAT-VERIFY.
00611    IF VALIDITY-INDICATOR = 'INVALID' GO TO READ-CARD-LOOP.
00612    IF CODE = 'R' OR 'F' OR 'A' OR 'C' PERFORM GENERATE-REPORTS, MOVE 'I' TO MATH-CHANGE-MADE.
00613    IF CODE = 'V' OR 'Q' OR 'E' OR 'M' OR 'X' OR 'D' PERFORM LOAD-MATH-CHANGE-FILE THRU END-LOAD-MATH-CHANGE-FILE.
00614    PERFORM MATH-LOAD-UPDATE.
00615    IF CODE = 'Z' OR 'A' OR 'U' OR 'M' OR 'X' OR 'D' PERFORM MATH-LOAD-UPDATE.
00616    MOVE 'Y' TO MATH-CHANGE-MADE.
00617    PERFORM LOAD-MATH-CHANGE-FILE THRU
00618    END-LOAD-MATH-CHANGE-FILE.
00619    PERFORM MATH-LOAD-UPDATE.
00620    GO TO READ-CARD-LOOP.
00621    CLOSE CARD-IN.
00622    STOP RUN.
00623    ACCEPT-DATF-FROM-SYSTEM.
00624    ACCEPT CURRENT-DAYT FROM DATE.
00625    MOVE DD OF CURRENT-DAYT TO DD OF DATE-IN-PRINT-FORM.
00626    MOVE YY OF CURRENT-DAYT TO YY OF DATE-IN-PRINT-FORM.
00627    MOVE MONTH-ALPHA (MM OF CURRENT-DAYT) TO MONTH OF DATE-IN-PRINT-FORM.
00628    MOVE DATE-IN-PRINT-FORM TO DATE-OUT OF DATE-HEAD.
00629    LOAD-CURRENT-QTR-DATA.
00630    IF CODE IS NOT EQUAL TO 'Y' THEN
00631    MOVE 'MISSING OR OUT OF SEQUENCE CURRENT QUARTER CARD OR PROCESS TERMINATING.' TO ERR-MSG-LINE
00632    PERFORM INVALID-INPUT.
00633    MOVE 'I' TO KODE.
00634    PERFORM LOAD-ALL-CURRENT-QTR-DATA.
00635    MOVE 'Y' TO YQ.
00636    GO TO END-CURRENT-QTR-DATA.
00637    SET YQ TO 1.
00638    MOVE 1 TO PIR.
00639    SEARCH YR-QTR VARYING PIR.
00640    AT END MOVE * INVALID CURRENT QUARTER SPECIFIED ON INPUT CARD = PROCESSING TERMINATING.' TO ERR-MSG-LINE
00641    PERFORM INVALID-INPUT.
00642    MOVE 'I' TO KODE.
00643    PERFORM LOAD-ALL-CURRENT-QTR-DATA.
00644    WHEN CURRENT-QTR IS EQUAL TO YR-QTR (YQ)
00645    MOVE PIR TO CURRENT-QTR-CODE.
00646    PERFORM LOAD-ALL-CURRENT-QTR-DATA.
00647    OPEN OUTPUT PRINT-FILE.
00648    MOVE 'Y' TO KODE.
00649    MOVE 'Y' TO KODE.
00650    PERFORM LOAD-ALL-CURRENT-QTR-DATA.
00651    WRITE PRINT-LINE FROM CURRENT-QTR-CODE AFTER POSITIONING 3.
LINES.
CLOSE PRINT-FILE.
END-CURRENT-QTR-DATA. EXIT.
INVALID-INPUT SECTION.

00660 OPEN OUTPUT PRINT-FILE.
00661 WRITE PRINT-LINE FROM ERROR-WARNING
  AFTER POSITIONING 3 LINES.
00662 WRITE PRINT-LINE FROM ERR-MSG-LINE
  AFTER POSITIONING 2 LINES.
00663 MOVE SPACES TO ERR-MSG-LINE.
00664 STRING 'THE FOLLOWING INPUT CARD DISREGARDED: ' CODE-CHECK
00665 DELIMITED BY SIZE
00666 INTO ERR-MSG-LINE.
00667 WRITE PRINT-LINE FROM ERR-MSG-LINE
00668 AFTER POSITIONING 2 LINES.
00669 MOVE SPACES TO ERR-MSG-LINE.
00670 CLOSE PRINT-FILE.
CARD-FORMAT-VERIFY SECTION.

REPORT-REQUEST.

MOVE 'VALID' TO VALIDITY-INDICATOR.

IF COOD = 'R' OR 'F' OR 'A' OR 'C' OR 'V' OR 'I' OR '7' OR 'MI' OR 'XI'

THEN IF PEST IS NOT EQUAL TO SPACES THEN

MOVE 'ILLEGAL ENTRY ON REPORT REQUEST' TO ERR-MSG-LINE

PERFORM INVALID-INPUT

MOVE 'INVALID' TO VALIDITY-INDICATOR.

IF COOD = 'V'

THEN IF HIST-DATA IS NOT NUMERIC OR FILLER-H IS NOT = ' ' 

MOVE 'ILLEGAL ENTRY ON HISTORY ALTERATION REQUEST

TO ERR-MSG-LINE

PERFORM INVALID-INPUT

MOVE 'INVALID' TO VALIDITY-INDICATOR.

IF COOD = 'V'

THEN IF DATA-Y IS NOT NUMERIC OR FILLER-Y IS NOT = ' '

MOVE 'ILLEGAL ENTRY ON YEAR AND QUARTER INPUT CARD' 

TO ERR-MSG-LINE

PERFORM INVALID-INPUT

MOVE 'INVALID' TO VALIDITY-INDICATOR.

IF COOD = 'V' OR '2' OR '3' OR '4' OR '5' OR '6' OR '7' OR '8' OR '9'

THEN IF SSN OF BASIC-DATA-1 IS NOT NUMERIC THEN

MOVE 'INVALID SSN ON INPUT CARD' TO ERR-MSG-LINE

PERFORM INVALID-INPUT

MOVE 'INVALID' TO VALIDITY-INDICATOR.

IF COOD = 'V'

THEN IF ADVISOR-NAME OF ADVISOR-V IS NOT ALPHABETIC 

OR ADVISOR-SSN OF ADVISOR-V IS NOT NUMERIC

MOVE 'ILLEGAL INPUT ON ADVISOR INPUT CARD' TO 

ERR-MSG-LINE

PERFORM INVALID-INPUT

MOVE 'INVALID' TO VALIDITY-INDICATOR.

IF COOD = 'V'

GO TO END-VERIFY.

IF COOD IS NOT EQUAL TO 'R' AND 'F' AND 'A' AND 'C' AND 'V' AND 'I' AND '7' AND 'MI' AND 'XI' THEN

MOVE 'INVALID' TO VALIDITY-INDICATOR

MOVE 'INVALID CODE IN COLUMN 1 OF INPUT CARD' TO 

ERR-MSG-LINE

PERFORM INVALID-INPUT.

END-VERIFY.
00717    LOAD=MATH-CHANGE=FILE SECTION.
00718    OPEN OUTPUT MATH-CHANGE.
00719    READ-INTO-MATH-CHANGE.
00720    MOVE SPACES TO CODE-CHECK=W.
00721    MOVE CODE-CHECK TO CODE-CHECK=W.
00722    WRITE CODE-CHECK=W.
00723    READ-CHANGE.
00724    READ CARD-IN AT END GO TO END-LOAD-MATH-CHANGE-FILE.
00725    PERFORM CARD-FORMAT-VERIFY.
00726    IF VALIDITY-INDICATOR = 'INVALID' GO TO READ-CHANGE.
00727    IF CODE IS EQUAL TO 'V' OR 'Q' OR 'Z' OR 'M' OR 'X' OR 'D'
00728    THEN GO TO READ-INTO-MATH-CHANGE.
00729    IF CODE IS NOT EQUAL TO 'X'
00730    THEN OPEN OUTPUT PRINT-FILE
00731    MOVE * EXPECTED TO FIND * AT END OF MATH CHANGE.
00732    = *CARD INPUT FOUND INSTEAD THE FOLLOWING DATA WHICH IS DISPF
00733    = *CARD INPUT FOUND INSTEAD THE FOLLOWING DATA WHICH IS DISPF
00734    *GARDED! * TO PRINT-LINE
00735    WRITE PRINT-LINE AFTER POSITIONING 2 LINES
00736    STRING * CARD IMAGE: * CODE-CHECK DELIMITED BY SIZE
00737    INTO PRINT-LINE
00738    WRITE PRINT-LINE AFTER POSITIONING 2 LINES
00739    CLOSE PRINT-FILE.
00740    CLOSE MATH-CHANGE.
00741    MOVE 102400 TO SORT-CORE-SIZE
00742    SORT INPUT-SORT-FILE
00743    ASCENDING KEY SSN OF INPUT-SORT
00744    USING MATH-CHANGE
00745    GIVING MATH-CHANGE.
00746    END-LOAD-MATH-CHANGE-FILE. EXIT.
00748  **** REFERENCE HIPO 2.1 ****
00749  MATH-LOAD-UPDATE SECTION.
00750  IF GOOD = 'T' THEN PERFORM STAT-HIST THRU EXIT-STAT-HIST.
00751  ELSE PERFORM MERGE=MATH-DATA THRU END-MERGE=MATH-DATA.
00752  GO TO END-MATH-LOAD-UPDATE-SECTION.

00754  MERGE=MATH-DATA.
00755  OPEN I-O MATH-DEPT-STUD-DR.
00756  OPEN INPUT MATH-CHANGE.
00757  OPEN OUTPUT MATH-REORG.
00758  MOVE SPACES TO MATH-REORG-REC.
00759  READ MATH-CHANGE AT END GO TO TRANSFER-REST-OF-MATH-FILE.
00760  READ MATH-DEPT-STUD-DR AT END GO TO CLOSE-MATH-FILES.
00761  PERFORM INITIALIZE-RECORD-BUFF.
00762  IF SSN OF MATH-CHANGE-REC < SSN OF STUD-REC-M
00763  THEN PERFORM CREATE=MATH-REC, PERFORM READ-MATH-CHANGE
00764  ELSE MOVE CORR STUD-REC-M TO RECORD-BUFF, PERFORM
00765  READ-FILES-TO-BE-MERGED.
00766  GO TO COMPARE-FOR-MERGE.

00767  READ-FILES-TO-BE-MERGED.
00768  READ MATH-DEPT-STUD-DR AT END
00770  PERFORM WRITE-NEW-FILES-ONLY
00771  GO TO CLOSE-MATH-FILES.
00772  READ-MATH-CHANGE.
00773  READ MATH-CHANGE AT END
00774  GO TO TRANSFER-REST-OF-MATH-FILE.
00775  COMPARE-FOR-MERGE.
00776  IF SSN OF MATH-CHANGE-REC = SSN OF RECORD-BUFF
00777  PERFORM UPDATE=MATH-REC, PERFORM READ-MATH-CHANGE
00779  GO TO COMPARE-FOR-MERGE.
00780  IF DELETE-CHAR IS NOT = '0'
00781  THEN IF RECORD-BUFF IS NOT = 0
00782  THEN WRITE MATH-REORG-REC FROM RECORD-BUFF.
00783  PERFORM INITIALIZE-RECORD-BUFF.
00784  ELSE NEXT SENTENCE.
00785  ELSE MOVE SSN OF RECORD-BUFF TO SSN OF MATH-REORG-REC
00786  PERFORM INITIALIZE-RECORD-BUFF.
00787  MOVE SSN OF MATH-REORG-REC TO SSN OF RECORD-BUFF.
00788  WRITE MATH-REORG-REC FROM RECORD-BUFF.
00789  MOVE SPACES TO DELETE-CHAR.
00790  IF SSN OF MATH-CHANGE-REC = SSN OF STUD-REC-M
00791  THEN MOVE CORR STUD-REC-M TO RECORD-BUFF.
00792  PERFORM UPDATE=MATH-REC.
00793  PERFORM READ-MATH-CHANGE.
00794  PERFORM READ-FILES-TO-BE-MERGED.
00795  ELSE IF SSN OF MATH-CHANGE-REC > SSN OF STUD-REC-M
00796  THEN MOVE STUD-REC-M TO RECORD-BUFF.
00797  WRITE MATH-REORG-REC FROM RECORD-BUFF.
00798  PERFORM INITIALIZE-RECORD-BUFF.
00799  PERFORM READ-FILES-TO-BE-MERGED.
00800  ELSE PERFORM CREATE=MATH-REC.
00801  PERFORM READ-MATH-CHANGE.
00802  GO TO COMPARE-FOR-MERGE.
21

00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457

INITIALIZE-RECORD-BUFF.
MOVE SPACES TO RECORD-BUFF, MOVE ZEROS TO SSN OF RECORD-BUFF, MOVE ZEROS TO ADVISOR-SSN OF RECORD-BUFF.
WRITE-NEW-FILES-ONLY.
IF SSN OF RECORD-BUFF IS EQUAL TO 0
THEN PERFORM CREATE-MATH-REC.
ELSE IF SSN OF RECORD-BUFF IS EQUAL TO SSN OF RUFF
THEN PERFORM CREATE-MATH-REC.
ELSE WRITE MATH-REORG-REC FROM RECORD-BUFF.
PERFORM INITIALIZE-RECORD-BUFF.
PERFORM CREATE-MATH-REC.
READ MATH-CHANGE AT END WRITE MATH-REORG-REC FROM RECORD-BUFF, GO TO CLOSE-MATH-FILES.
GO TO WRITE-NEW-FILES-ONLY.
TRANSFER-REST-OF-MATH-FILE.
IF SSN OF RECORD-BUFF IS NOT = 0
THEN IF SSN OF RECORD-BUFF = SSN OF STUD-REC-M
THEN WRITE MATH-REORG-REC FROM RECORD-BUFF
ELSE WRITE MATH-REORG-REC FROM RECORD-BUFF.
PERFORM INITIALIZE-RECORD-BUFF.
WRITE MATH-REORG-REC FROM STUD-REC-M.
ELSE WRITE MATH-REORG-REC FROM STUD-REC-M.
READ MATH-DEPT-STUD-DR AT END GO TO CLOSE-MATH-FILES.
GO TO TRANSFER-REST-OF-MATH-FILE.
CLOSE-MATH-FILES.
MOVE SPACES TO DELETE-CHAR.
CLOSE MATH-DEPT-STUD-DR.
MATH-CHANGE.
MATH-REORG.
OPEN INPUT MATH-REORG.
OUTPUT MATH-DEPT-STUD-DR.
TRANSFER-TO-MATH-DR.
READ MATH-REORG AT END GO TO END-MERGE-MATH-DATA.
MOVE SPACES TO STUD-REC-M.
MOVE ZEROS TO SSN OF STUD-REC-M, MOVE ZEROS TO ADVISOR-SSN OF STUD-REC-M.
IF SSN OF MATH-REORG-REC IS NOT EQUAL TO 0
THEN WRITE STUD-REC-M FROM MATH-REORG-REC.
END-MERGE-MATH-DATA.
CLOSE MATH-REORG, MATH-DEPT-STUD-DR.

***** REFERENCE HIPO 2.1.1 *****

00451
00452
00453
00454
00455
00456
00457

CREATE-MATH-REC.
IF COWD = 'V' MOVE CORR ADVISOR-W TO RECORD-BUFF.
ELSE IF COWD = 'D' MOVE CORR FTU-ATTEND-W TO RECORD-BUFF.
ELSE IF COWD = 'M' MOVE CORR ASSN-MBR-W
**TO RECORD-BUFF.**

ELSE

STRING 'CANNOT LOCATE RECORD TO BE CHANGED FOR SSN';

SSN OF MBR-DELETE-W DELIMITED BY SIZE INTO

ERR-MSG-LINE

MOVE MATH-CHANGE-REC TO CODE-CHECK

PERFORM INVALID-INPUT.

***REFERENCE HIPO 2.1.2***

UPDATE-MATH-REC.

IF COWD = 'V' PERFORM CODE-V THRU CODE-V-EXIT.

IF COWD = 'Q' PERFORM CODE-Q THRU CODE-Q-EXIT.

IF COWD = 'Z' PERFORM CODE-Z THRU CODE-Z-EXIT.

IF COWD = 'D' PERFORM CODE-D THRU CODE-D-EXIT.

CODE-V.

MOVE CORR ADVISOR-W TO RECORD-BUFF.

CODE-V-EXIT. EXIT.

CODE-Q.

MOVE SPACES TO WORK-AREA1.

MOVE SPACES TO WORK-AREA2.

MOVE ATTEND-DATA OF FTU-ATTEND-W TO WORK-AREA1.

MOVE ATTEND-DATA OF RECORD-BUFF TO WORK-AREA2.

MOVE SPACES TO ATTEND-DATA OF RECORD-BUFF.

STRING WORK-AREA2 WORK-AREA1 DELIMITED BY

' ' INTO ATTEND-DATA OF RECORD-BUFF.

CODE-Q-EXIT. EXIT.

CODE-W.

MOVE SPACES TO WORK-AREA1.

MOVE SPACES TO WORK-AREA2.

MOVE MBR-DATA OF ASSN-MBR-W TO WORK-AREA1.

MOVE MBR-DATA OF RECORD-BUFF TO WORK-AREA2.

MOVE SPACES TO MBR-DATA OF RECORD-BUFF.

STRING WORK-AREA2 WORK-AREA1 DELIMITED BY

' ' INTO MBR-DATA OF RECORD-BUFF.

CODE-W-EXIT. EXIT.

CODE-D.

MOVE '1' TO DELETE-CHAR.

CODE-D-EXIT. EXIT.

CODE-Z.

IF ATTEND-DATA OF FTU-ATTEND-DELETE-W IS EQUAL TO SPACES

THEN MOVE SPACES TO ATTEND-DATA OF RECORD-BUFF.

GO TO CODE-Z-EXIT.

MOVE SPACES TO ATTEND-ARRAY.

MOVE ATTEND-DATA OF RECORD-BUFF TO ATTEND-ARRAY.

MOVE 1 TO PTR.

SET 0 TO 1.

SEARCH PTR-ATTEND VARYING PTR AT END MOVE

* ATTENDANCE DATA ENTRY TO BE DELETED NOT FOUND * TO

ERR-MSG-LINE
PERFORM INVALID-INPUT
WHEN QTR-ATTEND (Q) ≠ ATTEND-DATA OF FTU-ATTEND-DELETE-W
MOVE ZEROS TO QTR-ATTEND (Q)
PERFORM FILL-IN-A UNTIL PTR = 30
MOVE ATTEND-ARRAY TO ATTEND-DATA OF RECORD-RUFF.
CODE-Z-EXIT. EXIT.

FILL-IN-A.
MOVE QTR-ATTEND (Q + 1) TO QTR-ATTEND (Q).
SET 0 UP BY 1.
ADD 1 TO PTR.
END-FILL-IN-A. EXIT.

CODE-X.
MOVE SPACES TO MRR-TAR-FIELD.
SET M TO 1.
MOVE 1 TO PTR.
UNSTRING MRR-DATA OF RECORD-RUFF DELIMITED BY '!' INTO
MRR (M) MRR (M + 1) MRR (M + 2) MRR (M + 3) MRR (M + 4)
MRR (M + 5) MRR (M + 6) MRR (M + 7) MRR (M + 8)
MRR (M + 9) MRR (M + 10).
SEARCH MRR VARYING PTR AT END MOVE
* MEMBER ENTRY TO BE DELETED NOT FOUND!
TO ERR-MSG-LINE PERFORM INVALID-INPUT
WHEN MRR (M) ≠ ASSN-NAME OF MRR-DELETE-W
MOVE SPACES TO MRR (M)
PERFORM FILL-IN-M UNTIL PTR = 9
MOVE MRR-TAR-FIELD TO MRR-DATA OF RECORD-RUFF.
CODE-X-EXIT. EXIT.

FILL-IN-M.
MOVE MRR (M + 1) TO MRR (M).
SET M UP BY 1.
ADD 1 TO PTR.
END-FILL-IN-M. EXIT.

***** REFERENCE HIPQ 2.1.3 *****
STAT-HIST.
OPEN 1-0 STAT-HIST-DB, OUTPUT HIST-REORG.
OUTPUT PRINT-FILE.
MOVE ZEROS TO HIST-REORG-REC.
IF CODE = 'H'
MOVE ZEROS TO ACCUMULATORS
MOVE CORR HIST-DATA TO ACCUMULATORS
MOVE QTR-H TO QTR-RUFF
PERFORM READ-HIST THRU EXIT-READ-HIST.
CLOSE STAT-HIST-DB, HIST-REORG, PRINT-FILE.
IF QTR-RUFF IS NOT EQUAL TO 0
THEN IF ACCUMULATORS IS EQUAL TO 0
THEN MOVE
' ENTRY TO BE DELETED NOT FOUND IN STATISTICS HIST FILE'
TO ERR-MSG-LINE
PERFORM INVALID-INPUT.
PERFORM SORT-STAT-HIST.
EXIT-STAT-HIST. EXIT.
READ-HIST.
READ STAT-HIST-DR AT END GO TO END-READ-HIST.

IF QTR-OF STAT-REC IS EQUAL TO QTR-RUFF,
THEN MOVE CORR-ACCUMULATORS TO STAT-REC,
MOVE ZEROS TO QTR-RUFF.

IF COMP-UP OF STAT-REC IS NOT EQUAL TO 0,
THEN WRITE HIST-REORG-REC FROM STAT-REC.

GO TO READ-HIST.

END-READ-HIST.

WRITE HIST-REORG-REC.

MOVE ZEROS TO HIST-REORG-REC.

SORT-STAT-HIST.

MOVE 102400 TO SORT-CORF-SIZE.

SORT SORT-WORK-HDR

DESCENDING KEY QTR OF SORT-HDR-REC

USING HIST-REORG GIVING STAT-HIST-DR.

END-MATH-LOAD-UPDATE-SECTION.
00996  ***** REFERENCE HIPO 2.2 *****
00997  LOAD-FTU-DATA SECTION.
00998  IF CODD IS NOT EQUAL TO '1' AND '2' AND '3' AND '4' THEN
00999    GO TO GET-SPIN-OFF.
01000    OPEN OUTPUT SPIN-OFF.
01001    INIT-SPIN-REC.
01002    MOVE SPACES TO SPIN-OFF-REC.
01003    MOVE ZEROS TO SSN OF SPIN-OFF-REC.
01004    MOVE ZEROS TO FTU-SUMMARY OF SPIN-OFF-REC.
01005    **** REFERENCE HIPO 2.2.2 *****
01006    GET-SPIN-OFF.
01007    IF CODD = '1' MOVE CORP BASIC-DATA-1 TO SPIN-OFF-REC,
01008    IF CODD = '2' MOVE CORP STUD-ADDRESS-2 TO SPIN-OFF-REC,
01009    IF CODD = '3' MOVE CORP PARENT-ADDRESS-3 TO SPIN-OFF-REC.
01010    IF CODD = '4' MOVE CORP INDEX-4 TO INDEX-CONVERT
01011    MOVE INDEX-CONVERT TO INDX OF SPIN-OFF-REC
01012    MOVE SSN OF INDEX-4 TO SSN OF SPIN-OFF-REC.
01013    END-DECODE. EXIT.
01014    READ-NEXT-CARD.
01015    READ CARD IN AT END GO TO
01016    CHECK-FOR-END-OF-SPO.
01017    IF CODD IS NOT EQUAL TO '1' AND '2' AND '3' AND '4' THEN
01018    WRITE SPIN-OFF-REC
01019    CLOSE SPIN-OFF.
01020    GO TO CHECK-FOR-END-OF-SPO.
01021    IF SSN OF BASIC-DATA-1 IS NOT EQUAL TO SSN OF SPIN-OFF-REC
01022    THEN WRITE SPIN-OFF-REC
01023    PERFORM INITIALIZE-SPIN-OFF.
01024    PERFORM DECODE-SPIN-OFF THRU END-DECODE.
01025    GO TO READ-NEXT-CARD.
01026    **** REFERENCE HIPO 2.2.1 *****
01027    GET-SPIN-OFF.
01028    * GET-SPIN-OFF TO BE WRITTEN.
01029
01030    CHECK-FOR-END-OF-SPO.
01031    IF CODD IS NOT EQUAL TO '1' THEN
01032    WRITE 'R FOR SPIN-OFF' TO ERR-MSG-LINE.
01033    PERFORM INVALID-INPUT.
01034
01035    **** REFERENCE HIPO 2.2.3 *****
01036    LOAD-STUD-DR.
01037    OPEN INPUT SPIN-OFF.
01038    OPEN OUTPUT STUD-DATA-REC.
01039    PERFORM INITIALIZE-STUD-DR-REC.
01040    TRANSFER-DATA.
01041    READ SPIN-OFF AT END GO TO END-OF-LOAD-STUD-DR.
01042    MOVE CORP SPIN-OFF-REC TO STUD-DR-REC.
01043    MOVE FTU-GPA OF SPIN-OFF-REC TO GPA OF STUD-DR-REC.
01050  MOST-RECENT-QTR-AND-IF-GRAD.
01051  MOVE INDX OF SPIN-OFF-REC TO INDEX-ARRAY-LOAD-FIELD.
01052  MOVE 0 TO CTR.
01053  SET-CTR-LOOP.
01054  MOVE CURRENT-QTR-CODE TO SUBSCRPT, SUBTRACT CTR FROM
01055  SUBSCRPT.
01056  IF INDEX-ENTRY (SUBSCRPT)
01057      IS NOT EQUAL TO 000000 THEN
01058      GO TO HAS-STUD-GRADUATED.
01059      ADD 1 TO CTR.
01060      IF (CURRENT-QTR-CODE = CTR) > 0 GO TO SET-CTR-LOOP
01061      ELSE STRING * * CANNOT DETERMINE LAST QTR ATTENDED FOR
01062      LAST-NAME OF SPIN-OFF-REC-FRST-MDL-NAME OF SPIN-OFF-REC
01063      1 CHECK SPIN-OFF DATA AND CURRENT QTR INPUT.
01064      DELIMITED BY SIZE INTO ERR-MSG-LINE
01065      OPEN OUTPUT PRINT-FILE
01066      WRITE PRINT-LINE FROM ERR-MSG-LINE AFTER POSITIONING
01067      3 LINES
01068      CLOSE PRINT-FILE
01069      MOVE SPACES TO ERR-MSG-LINE.
01070      IF (CURRENT-QTR-CODE = CTR) = 0
01071      THEN MOVE 'I' TO GRAD OF STUD-DR-REC
01072      MOVE ZEROS TO F'TU-LAST-QTR OF STUD-DR-REC
01073      GO TO STUD-LEVEL-OF-STUDY
01074      MOVE CURRENT-QTR-CODE TO SUBSCRPT, SUBTRACT CTR FROM
01075      SUBSCRPT.
01076      IF GRAD-IND OF INDEX-ENTRY (SUBSCRPT)
01077      IS EQUAL TO '1' OR '2'
01078      MOVE 'Y' TO GRAD STU OF STUD-DR-REC
01079      ELSE MOVE 'N' TO GRAD STU OF STUD-DR-REC.
01080      SET YQ TO SUBSCRPT
01081      MOVE YR-QTR (YQ) TO FTU-LAST-QTR OF STUD-DR-REC.
01082      STUD-LEVEL-OF-STUDY.
01083      IF APPL-TYPE OF SPIN-OFF-REC < 5 MOVE 'U' TO
01084      GRAD-UG-PR OF STUD-DR-REC.
01085      IF APPL-TYPE OF SPIN-OFF-REC = 5 MOVE 'P' TO
01086      GRAD-UG-PR OF STUD-DR-REC.
01087      IF APPL-TYPE OF SPIN-OFF-REC = 6 OR APPL-TYPE OF SPIN-OFF-REC
01088      = 8 MOVE 'G' TO GRAD-UG-PR OF STUD-DR-REC.
01089      IF APPL-TYPE OF SPIN-OFF-REC IS NOT EQUAL TO 1 AND 2 AND 3
01090      AND 4 AND 5 AND 6 AND 8 THEN
01091      MOVE 'I' TO GRAD-UG-PR OF STUD-DR-REC
01092      STRING ' * CANNOT DETERMINE IF ' LAST-NAME OF
01093      SPIN-OFF-REC-FRST-MDL-NAME OF SPIN-OFF-REC
01094      SSN OF SPIN-OFF-REC
01095      IS GRAD-UG-PR OR PR - IS NOT INCLUDED IN STATISTICS
01096      - ' ' DELIMITED BY SIZE INTO ERR-MSG-LINE
01097      OPEN OUTPUT PRINT-FILE
01098      WRITE PRINT-LINE FROM ERR-MSG-LINE AFTER POSITIONING
01099      3 LINES
01100      CLOSE PRINT-FILE
01101      MOVE SPACES TO ERR-MSG-LINE.
01102      IF MAJOR-1 OF SPIN-OFF-REC = 1701
01103      ENTER-STUD-MAJOR.
01107 THEN MOVE 'MATH' TO MAJOR OF STUD-DB-REC.
01108 IF MAJOR-1 OF SPIN-OFF-REC = 1790
01109 THEN MOVE 'MATH' TO MAJOR OF STUD-DB-REC.
01110 IF MAJOR-1 OF SPIN-OFF-REC = 0701
01111 THEN MOVE 'COMP' TO MAJOR OF STUD-DB-REC.
01112 IF MAJOR-1 OF SPIN-OFF-REC = 1702
01113 THEN MOVE 'STAT' TO MAJOR OF STUD-DB-REC.
01114 IF MAJOR-1 OF SPIN-OFF-REC IS NOT EQUAL TO
01115 1701 AND 1790 AND 0701 AND 1702
01116 THEN MOVE '????' TO MAJOR OF STUD-DB-REC
01117 STRING * CANNOT DETERMINE MAJOR OF * LAST-NAME OF
01118 SPIN-OFF-REC FIRST-NAME OF SPIN-OFF-REC SSN OF
01119 SPIN-OFF-REC
01120 * DATA NOT INCLUDED IN STATISTICS' DELIMITED BY SIZE
01121 INTO ERR-MSG-LINE
01122 OPEN OUTPUT PRINT-FILE
01123 WRITE PRINT-LINE FROM ERR-MSG-LINE AFTER POSITIONING
01124 3 LINES
01125 CLOSE PRINT-FILE
01126 MOVE SPACES TO ERR-MSG-LINE.
01127 MOVE-GPA.
01128 MOVE FTU-GPA OF FTU-SUMMARY OF SPIN-OFF-REC TO GPA OF
01129 STUD-DB-REC.
01130 WRITE-STUD-DR-REC.
01131 IF SSN OF STUD-DB-REC IS NOT EQUAL TO 0
01132 THEN IF SSN OF STUD-DR-REC IS NUMERIC
01133 THEN WRITE STUD-DR-REC.
01134 PERFORM INITIALIZE-STUD-DR-REC.
01135 GO TO TRANSFER-DATA.
01136 INITIALIZE-STUD-DR-REC.
01137 MOVE SPACES TO STUD-DR-REC.
01138 MOVE ZEROS TO GPA OF STUD-DB-REC.
01139 MOVE ZEROS TO FTU-LAST-TR OF STUD-DR-REC.
01140 MOVE ZEROS TO ADVISOR-SSN OF STUD-DB-REC.
01141 MOVE ZEROS TO SSN OF STUD-DB-REC.
01142 END-OF-LOAD-STUD-DR.
01143 CLOSE SPIN-OFF, STUD-DATA-BASE.
01144 END-FTU-DATA-SECTION.
***** REFERENCE HIPO 2.3 *****

**GENERATE-REPORTS SECTION.**

SORT SORT-WORK-SDR

ASCENDING KEY SSN OF SORT-SDR-REC

USING STUD-DATA-BASE

GIVING STUD-DATA-RAF.

IF MATH-CHANGE-MADE = 'N' GO TO BRANCH-TO-PRINT-RTN.

OPEN OUTPUT PRINT-FILE.

OPEN INPUT MATH-DEPT-STUD-DR.

OPEN 1-O STUD-DATA-RAF.

PERFORM MERGE-DATA.

PERFORM READ-NEXT-STUD-DR-REC.

PERFORM COMPARE-RECORDS THRU END-OF-MERGE-DATA.

CLOSE PRINT-FILE.

CLOSE MATH-DEPT-STUD-DR, STUD-DATA-RAF.

BRANCH-TO-PRINT-RTN.

PERFORM IDENTIFY-TYPE-REPORT THRU EXIT-IN-REPORT.

GO TO EXIT-GEN-RPTS-SECTION.

***** REFERENCE HIPO 2.3.1 *****

**MERGE-DATA.**

READ MATH-DEPT-STUD-DR AT END WRITE STUD-DR-REC.

GO TO END-OF-MERGE-DATE.

READ-NEXT-STUD-DR-REC.

READ STUD-DATA-RAF AT END GO TO END-OF-MERGE-DATE.

COMPARE-RECORDS.

IF SSN OF STUD-DR-REC = SSN OF STUD-REC-M THEN

MOVE CORR STUD-REC-M TO STUD-DR-REC

WRITE STUD-DR-REC PERFORM NULL-STUD-DR-REC

PERFORM READ-NEXT-STUD-DR-REC, PERFORM MERGE-DATA.

GO TO COMPARE-RECORDS.

IF SSN OF STUD-REC-M < SSN OF STUD-DR-REC THEN

STRING * 'SSN OF STUD-REC-M NOT FOUND IN FTU'SUPPLIED DATA, VERIFY SSN,' DELIMITED BY SIZE

INTO ERR-MSG-LINE

WRITE PRINT-LINE FROM ERR-MSG-LINE AFTER POSITIONING 3 LINES.

MOVE SPACES TO ERR-MSG-LINE.

PERFORM MERGE-DATA.

GO TO COMPARE-RECORDS.

WRITE STUD-DR-REC.

PERFORM READ-NEXT-STUD-DR-REC.

GO TO COMPARE-RECORDS.

END-OF-MERGE-DATE. EXIT.

NULL-STUD-DR-REC.

IF ADVISOR-NAME OF STUD-REC-M IS EQUAL TO SPACES

THEN IF ADVISOR-SSN OF STUD-REC-M IS EQUAL TO ZEROS

THEN IF ATTEND-DATA OF STUD-REC-M = SPACES

THEN IF MRP-DATA OF STUD-REC-M = SPACES

THEN MOVE ZEROS TO SSN OF STUD-REC-M.

***** REFERENCE HIPO 2.3 (CONT) *****

**IDENTIFY-TYPE-REPORT.**
IF COOD = 'C' PERFORM PRINT-STAT THRU END-PRINT-STAT.

IF COOD = 'R' OR 'F' THEN
MOVE 102400 TO SORT-CORE-SIZE

SORT SORT-WORK-SDR

DESCENDING KEY FTU-LAST-QTR OF SORT-SDR-REC
ASCENDING KEY MAJOR OF SORT-SDR-REC
ASCENDING KEY GRAD-UG-PR OF SORT-SDR-REC
ASCENDING KEY LAST-NAME OF SORT-SDR-REC
ASCENDING KEY FIRST-MDLE-NAME OF SORT-SDR-REC
USING STUD-DATA-RAISE

PERFORM PRINT-REPT THRU END-OF-PRINT-REPT.

IF COOD = 'A' THEN
MOVE 102400 TO SORT-CORE-SIZE

SORT SORT-WORK-SDR

ASCENDING KEY ADVISOR-SSN OF SORT-SDR-REC
ASCENDING KEY LAST-NAME OF SORT-SDR-REC
ASCENDING KEY FIRST-MDLE-NAME OF SORT-SDR-REC
USING STUD-DATA-RAISE

PERFORM PRINT-REPT THRU END-OF-PRINT-REPT.

EXIT-IN-REPORT. EXIT.

**** REFERENCE HIPO 2.3.2 ****

PRINT-STAT.
OPEN OUTPUT PRINT-FILE.
MOVE ZEROS TO ACCUMULATORS.
MOVE ZEROS TO QT-BUFF.
WRITE PRINT-LINE FROM PAGE-HEAD-DEPT AFTER POSITIONING 0 LINES.
WRITE PRINT-LINE FROM STUD-STAT-HEAD AFTER POSITIONING 2 LINES.
OPEN INPUT STUD-DATA-BASE.

READ STUD-DATA-BASE AT END CLOSE STUD-DATA-BASE.

IF FTU-LAST-QTR OF STUD-DR-REC IS NOT EQUAL TO CURRENT-QTR THEN GO TO COMPUTE-CURRENT-STAT.

IF MAJOR OF STUD-DR-REC = 'COMP' THEN PERFORM COMP-TALLY.
IF MAJOR OF STUD-DR-REC = 'MATH' THEN PERFORM MATH-TALLY.
IF MAJOR OF STUD-DR-REC = 'STAT' THEN ADD 1 TO STAT-UUG.

GO TO COMPUTE-CURRENT-STAT.

COMP-TALLY.
IF GRAD-UG-PR OF STUD-DR-REC = 'G' THEN ADD 1 TO COMP-GP OF ACCUMULATORS.
IF GRAD-UG-PR OF STUD-DR-REC = 'U' THEN ADD 1 TO COMP-UG OF ACCUMULATORS.
IF GRAD-UG-PR OF STUD-DR-REC = 'P' THEN ADD 1 TO COMP-PR OF ACCUMULATORS.

MATH-TALLY.
IF GRAD-UG-PR OF STUD-DR-REC = 'G' THEN ADD 1 TO MATH-GP OF ACCUMULATORS.
IF GRAD-UG-PR OF STUD-DR-REC = 'U' THEN ADD 1 TO MATH-UG OF ACCUMULATORS.
IF GRAD-UG-PH OF STUD-DR-REC = 'K' THEN
  ADD 1 TO MATH-PR OF ACCUMULATORS.
  IF CURRENT-OHR-TO-OHR-RUFF = 'P'
    THEN MOVE CURRENT-OHR TO OHR-RUFF.
  MOVE CURRENT-OHR TO OHR-RUFF.
END-PRINT-STAT-LOOP.

READ STAT-HIST-DR AT END GO TO END-PRINT-STAT-LOOP.

MOVE ZEROS TO ACCUMULATORS.
MOVE CORP STAT-REC TO ACCUMULATORS.
ADD COMP-UG OF ACCUMULATORS COMP-GR OF ACCUMULATORS
  COMP-PR OF ACCUMULATORS GIVING COMPTOT OF ACCUMULATORS.
ADD MATH-UG OF ACCUMULATORS MATH-GR OF ACCUMULATORS
  MATH-PB OF ACCUMULATORS GIVING MATHTOT OF ACCUMULATORS.
ADD CORP ACCUMULATORS TO SLATOT OF ACCUMULATORS.
ADD COMPTOT OF ACCUMULATORS COMP-GR OF ACCUMULATORS
  STATOT OF ACCUMULATORS GIVING TOTSTSUNS OF ACCUMULATORS.
MOVE QTR OF STAT-HIST-DR TO QTR-PRINT.
WRITE PRINT-LINE FROM CURRENT-OHR-PRINT AFTER POSITIONING
  3 LINES.
WRITE PRINT-LINE FROM STUD-STAT-COL-HEAD AFTER POSITIONING
  2 LINES.
ADD MATH-GR OF ACCUMULATORS GIVING TOT-UG OF STUD-STAT-UGRAD.
MOVE CORP ACCUMULATORS TO STUD-STAT-UGRAD.
WRITE PRINT-LINE FROM STUD-STAT-UGRAD AFTER POSITIONING
  2 LINES.
ADD MATH-GR OF ACCUMULATORS GIVING TOT-UG OF STUD-STAT-GRAD.
MOVE CORP ACCUMULATORS TO STUD-STAT-GRAD.
WRITE PRINT-LINE FROM STUD-STAT-GRAD AFTER POSITIONING
  2 LINES.
ADD COMP-PR OF ACCUMULATORS MATH-PB OF ACCUMULATORS
  GIVING TOT-PR OF STUD-STAT-POST-RAC.
MOVE CORP ACCUMULATORS TO STUD-STAT-POST-RAC.
WRITE PRINT-LINE FROM STUD-STAT-POST-RAC AFTER
  POSITIONING 2 LINES.
MOVE CORP ACCUMULATORS TO STUD-STAT-TOT.
WRITE PRINT-LINE FROM STUD-STAT-TOT AFTER POSITIONING
  2 LINES.
MOVE SPACES TO PRINT-LINE.
WRITE PRINT-LINE AFTER POSITIONING 3 LINES.
GO TO PRINT-STAT-LOOP.
END-PRINT-STAT-LOOP.
MOVE SPACES TO PRINT-LINE.
WRITE PRINT-LINE AFTER POSITIONING 0 LINES.
CLOSE PRINT-FILE, STAT-HIST-DR.
END-PRINT-STAT, EXIT.

**** REFERENCE HIPO 2.3.3 ****
PRINT-REPT.
IF COOD = 'R' THEN MOVE 'R' TO KODE.
01313 IF C000 = 'F' THEN MOVE 'F' TO KORE.
01314 IF C000 = 'A' THEN MOVE 'A' TO KORE.
01315 PERFORM PRINT-RTN THRU PRINT-RTN-END.
01316 IF C000 = 'R' THEN
01317 MOVE 102400 TO SORT-CORE-SIZE
01318 SORT SORT-WORK-SDR
01319 DESCENDING KEY ADVISOR-SSN OF SORT-SDR-REC
01320 ASCENDING KEY LAST-NAME OF SORT-SDR-REC
01321 ASCENDING KEY FIRST-NAME OF SORT-SDR-REC
01322 USING STUD-DATA-RAISE
01323 GIVING STUD-DATA-RAISE
01324 MOVE 'S' TO KORE
01325 PERFORM PRINT-RTN THRU PRINT-RTN-END.
01326 END-OF-PRINT-RTN, EXIT.
01327 EXIT-GEN-RTS-SECTION, EXIT.
**** REFERENCE HIPO 4.1 ****

01329 PRINT-RTN SECTION.
01330 MOVE 'NO' TO BREAK-STATUS.
01331 OPEN INPUT STUD-DATA-BASE OUTPUT PRINT-FILE.
01332 HEAD STUD-DATA-BASE
01333 AT END MOVE '*' NO INPUT FROM STUD DATA BASE TO
01334 PRINT-LINE, WRITE PRINT-LINE AFTER POSITIONING 2 LINES
01335 GO TO EXIT-PRINT-RTN-SECTION.
01336 LOOP-PRINT-RTN:
01338 PERFORM LOAD-BREAK-FILEDS.
01339 PERFORM PRINT-HEADINGS THRU EXIT-PRINT-HEADINGS.
01340 PERFORM PRINT-DATA THRU EXIT-PRINT-DATA UNTIL BREAK-STATUS
01341 IS NOT EQUAL TO 'NO'.
01342 IF BREAK-STATUS IS NOT EQUAL TO 'EOF' THEN
01343 MOVE 'NO' TO BREAK-STATUS
01344 GO TO LOOP-PRINT-RTN.
01345 PRINT-RTN-END:
01346 MOVE SPACES TO PRINT-LINE.
01347 WRITE PRINT-LINE AFTER POSITIONING 0 LINES.
01348 CLOSE STUD-DATA-BASE, PRINT-FILE.
01349 EXIT-PRINT-RTN-SECTION.
01350 GO TO END-OF-PRINT-RTN-SECTION.

PRINT-HEADINGS:
01351 MOVE SPACES TO PRINT-LINE.
01352 WRITE PRINT-LINE FROM PAGE-HEAD-DEPT AFTER POSITIONING
01353 0 LINES.
01354 IF KODE = '0' OR '1' THEN
01355 PERFORM FORMULATE-SUR-HEADINGS THRU EXIT-SUR-HEAD.
01356 IF KODE = '7' OR '8'
01357 WRITE PRINT-LINE FROM FAC-ADV-HEAD AFTER POSITIONING
01358 2 LINES.
01359 IF KODE = '0' OR '1'
01360 WRITE PRINT-LINE FROM TITLE-ROST AFTER POSITIONING
01361 2 LINES.
01362 IF KODE = '5'
01363 MOVE 'FACULTY ADVISOR EXTRACT OF STUDENT ROST'
01364 WRITE TO TITLE-LINE
01365 WRITE PRINT-LINE FROM TITLE-ROST AFTER POSITIONING 2
01366 LINES.
01367 IF KODE = '0'
01368 WRITE PRINT-LINE FROM FORMER-STUD-HEAD AFTER
01369 POSITIONING 2 LINES.
01370 WRITE PRINT-LINE FROM DATE-HEAD AFTER POSITIONING 2 LINES.
01371 MOVE SPACES TO PRINT-LINE, WRITE PRINT-LINE AFTER POSITIONING
01372 2 LINES.
01373 IF KODE = 'A'
01374 THEN IF ADVISOR-NAME OF STUD-DR-REC IS EQUAL TO SPACES
01375 WRITE PRINT-LINE FROM ADVISEE-HEAD-ALT AFTER
01376 POSITIONING 2 LINES
01377 FLSE MOVE ADVISOR-NAME OF STUD-DR-REC TO ADVISOR-NAME
01378 OF ADVISEE-HEAD
01379 WRITE PRINT-LINE FROM ADVISEE-HEAD AFTER POSITIONING
01380 2 LINES.
01381 IF KODE = 'F'
01382 WRITE PRINT-LINE FROM FAC-ADV-COL-HEAD AFTER
01383 POSITIONING 2 LINES.
01384 EXIT-PRINT-HEADINGS. EXIT.
LOAD-BREAK-FIELDS.
MOVE MAJOR OF STUD-DR-REC TO BREAK-1.
MOVE GRAD-UG-PR OF STUD-DR-REC TO BREAK-2.
MOVE ADVISOR-SSN OF STUD-DR-REC TO BREAK-3.

FORMULATE-SUR-HEADINGS.
IF MAJOR OF STUD-DR-REC = 'COMPT' THEN
  MOVE 'GRADUATE COMPUTER SCIENCE MAJORS'
  TO TITLE-LINE
ELSE IF GRAD-UG-PR OF STUD-DR-REC = 'U' THEN
  MOVE 'UNDERGRADUATE COMPUTER SCIENCE MAJORS'
  TO TITLE-LINE
ELSE MOVE 'POST-HACALUAREATE COMPUTER SCIENCE MAJORS'
  TO TITLE-LINE
ELSE NEXT SENTENCE.

IF MAJOR OF STUD-DR-REC = 'MATH' THEN
  MOVE 'GRADUATE MATHEMATICAL SCIENCE MAJORS'
  TO TITLE-LINE
ELSE IF GRAD-UG-PR OF STUD-DR-REC = 'U' THEN
  MOVE 'UNDERGRADUATE MATHEMATICS MAJORS'
  TO TITLE-LINE
ELSE MOVE 'POST-HACALUAREATE MATHEMATICAL SCIENCE MAJORS'
  TO TITLE-LINE
ELSE NEXT SENTENCE.

IF MAJOR OF STUD-DR-REC = 'STAT' THEN
  MOVE 'UNDERGRADUATE STATISTICS MAJORS'
  TO TITLE-LINE.
EXIT-SUR-HEAD. EXIT.

PRINT-DATA.
MOVE CORP STUD-DR-REC TO ARR-NAMF.

IF KODE = 'S'
  THEN IF FTU-LAST-QTR OF STUD-DR-REC IS NOT EQUAL TO CURRENT-QTR
   THEN IF ADVISOR-SSN OF STUD-DR-REC IS EQUAL TO ZERO
    THEN PERFORM CONTINUE-ON, GO TO PRINT-DATA.

IF KODE = 'M'
  THEN IF FTU-LAST-QTR OF STUD-DR-REC IS NOT EQUAL TO CURRENT-QTR
   THEN PERFORM ROSTER-PRINT.

IF KODE = 'O'
  THEN IF FTU-LAST-QTR OF STUD-DR-REC IS NOT EQUAL TO CURRENT-QTR
   THEN IF ATTEND-DATE OF STUD-DR-REC IS EQUAL TO ' '
    THEN IF ADVISOR-NAME OF STUD-DR-REC IS NOT EQUAL TO ' '
     THEN IF ADVISOR-SSN OF STUD-DR-REC IS NOT EQUAL TO ' '
      THEN GO TO CONTINUE-ON
      ELSE PERFORM ROSTER-PRINT.

IF KODE = 'F' AND FTU-LAST-QTR OF STUD-DR-REC IS NOT CURRENT-QTR
  MOVE ARR-NAMF TO NAME OF FAC-ADV-NAMF.
  MOVE ADVISOR-NAME OF STUD-DR-REC TO ADVISOR-NAMF.

EXIT-PRINT-DATA.
01442 WRITE PRINT- LINE FROM FAC-ADV-DET- LINE AFTER POSITIONING 2 LINES.
01443
01444 IF KODE = 'TA'
01445 THEN IF FTU-LAST- QTR OF STUD- OR- REC IS EQUAL TO CURRENT- QTR.
01446
01447 MOVE ABBR- NAME TO NAME OF ADVISEE- DET- LINE.
01448 WRITE PRINT- LINE FROM ADVISEE- DET- LINE AFTER POSITIONING 2 LINES.
01449
CONTINUE- ON.
01450 READ STUD- DATA- BASE AT END MOVE 'EOF' TO BREAK- STATUS
01451
CONTINUE- ON- END.
01452
CONTINUE- ON- END.
01453
IF KODE = 'If'
01454 THEN IF FTU-LAST- QTR OF STUD- OR- REC IS NOT EQUAL TO CURRENT- QTR.
01455
GO TO EXIT- PRINT- DATA.
01456
IF KODE = 'Fr'
01457 THEN IF FTU-LAST- QTR OF STUD- OR- REC IS NOT EQUAL TO CURRENT- QTR.
01458 THEN MOVE '01' TO KODE.
01459
MOVE 'YES' TO BREAK- STATUS.
01460
GO TO EXIT- PRINT- DATA.
01461 IF KODE = 'Ti' OR 'Ie'
01462 THEN IF MAJOR OF STUD- OR- REC
01463 IS NOT EQUAL TO BREAK- 1 OR GRAD- UG- PR OF STUD- OR- REC
01464 THEN MOVE 'YES' TO BREAK- STATUS.
01465
IF KODE = 'At OR 'Is'
01466 THEN IF BREAK- 3 IS NOT EQUAL TO ADVISOR- SSN
01467 OF STUD- OR- REC THEN MOVE 'YES' TO BREAK- STATUS.
01468
IF KODE = '01' MOVE 'NO' TO BREAK- STATUS.
01469 EXIT- PRINT- DATA. EXIT.
01470

ROSTER- PRINT.
01471
MOVE CORP- STUD- OR- REC TO NAME- LINE.
01472
MOVE FTU-LAST- QTR OF STUD- OR- REC TO FTU-LAST- QTR- EDIT.
01473
MOVE LAST- YR- ATT- EDIT TO LAST- YR- ATT OF FTU-LAST- QTR
01474
OF NAME- LINE.
01475 IF LAST- YR- ATT = '11' THEN
01476
MOVE '11' TO LAST- QTR- ATT SUBTRACT 1 FROM LAST- YR- ATT- EDIT
01477
MOVE LAST- YR- ATT- EDIT TO LAST- YR- ATT.
01478
IF LAST- QTR- ATT = '12' THEN
01479
MOVE '12' TO LAST- QTR- ATT.
01480
IF LAST- QTR- ATT = '13' THEN
01481
MOVE '13' TO LAST- QTR- ATT.
01482 IF LAST- QTR- ATT = '14' THEN
01483
MOVE '14' TO LAST- QTR- ATT.
01484
IF MAJOR- STATUS OF NAME- LINE = '11' THEN
01485
MOVE '11' TO MAJOR- STATUS OF NAME- LINE.
01486
IF MAJOR- STATUS OF NAME- LINE = '12' THEN
01487
MOVE '12' TO MAJOR- STATUS OF NAME- LINE.
01488
IF MAJOR- STATUS OF NAME- LINE = '13' THEN
01489
MOVE '13' TO MAJOR- STATUS OF NAME- LINE.
01490
WRITE PRINT- LINE FROM NAME- LINE AFTER POSITIONING 2 LINES.
01491
MOVE CORP- STUD- OR- REC TO ADDR- LINE.
01492
WRITE PRINT- LINE FROM ADDR- LINE AFTER POSITIONING 1 LINES.
MOVE CORR STUD-DR-REC TO ADDR-PARENT-LINE.
WRITE PRINT-LINE FROM ADDR-PARENT-LINE AFTER POSITIONING 1 LINES.
MOVE CORR STUD-DR-REC TO FTU-ATTN-LINE.
WRITE PRINT-LINE FROM FTU-ATTN-LINE AFTER POSITIONING 1 LINES.
END-OF-PRINT-RTN-SECTION.