UNIVERSITY SPACE PLANNING:

PROJECTIONS FOR KANSAS STATE UNIVERSITY

by

K. CHANDRASHEKAR

B. E. (Mechanical Engineering)

University of Madras, India, 1970

A MASTER'S REPORT

submitted in partial fulfillment of the

requirement for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1973

Approved by:

L. E. Grosh

Major Professor

To my parents

ACKNOWLEDGEMENTS

I am grateful to my Major Professor Dr. L. E. Grosh for his encouragement and valuable guidance through this work. I am also grateful to Dr. Cecil Best, Dr. Doris Grosh and Dr. Frank Tillman for their comments on the work.

TABLE OF CONTENTS

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

CHAPTER 1

PROBLEM STATEMENT AND DISCUSSION OF LITERATURE

1.1 INTRODUCTION.

Millions of federal, state and local dollars are being spent annually by colleges and universities throughout the United States for new physical facilities. As the cost of providing facilities has become more burdensome, the need for effective planning and utilization of these facilities has become a major concern of not only institutional administrators, but also of those called upon to provide the capital funds. The importance of this problem was realized by the Board of Regents, State of Kansas, the governing body of the state colleges and universities. They began the massive task of developing a comprehensive master academic plan in conjunction with a long range physical development plan. A private consulting firm, Claudill, Rowlett and Scott (herto referred as CRS), was hired by the Board of Regents to provide a physical facilities projection scheme for all six colleges and universities under its jurisdiction. CRS delivered a master plan and presented their space planning reports in March 1972. Extensive review and evaluation of these reports indicated that they are adequate for comparing overall effort of the institutions, but lacking in clarity or contrast between organizational units, within and among the six schools. The work undertaken here concerns itself with refining the previous reports.

1.2 LITERATURE.

Several groups have worked in the general area of facilities planning. The sections which follow identify these groups and discuss the work they have done in relation to facilities planning.

1.2.1 BACKGROUND MATERIAL.

Background material is provided by the papers presented at the symposium of Operations Analysis of Education (2). These papers discuss the broad field of planning in a university, and do not deal with any particular aspect in detail. These papers have interesting information and give a good idea of the scope of the thinking in this field. They are, however, not directly related to the subject of facilities planning.

1.2.2 THE TORONTO GROUP.

Judy and Levine (4) and Van Vijk, Russel and Atcheson (5) discuss their development of a simulation model for a university. Their effort was given the acronym CAMPUS (Comprehensive Analytical Model for Planning in the University Sphere). They simulated university operations over a time period of any length. Their model "..... Loaded into the computer, accepts descriptions of the university's structure and statements of the levels of activities that the university is expected to perform. With these inputs, the model computes the resulting resource requirements of staff, space, materials and money." (4, p. vii). The model is divided into four main sections. The first section called 'Enrollment Formulation' receives the input data on projected enrollment for each year of the simulated period and from this it distributes the enrollees among the various colleges and thence into academic courses of study. From this it goes on to compute the subject enrollees for each department. The second section called 'Resource Loading' accepts as input the number of subject enrollees requiring instruction in each department. The third section called 'Space Requirements' accepts as input the output from the previous section and computes the space requirements, using space factors, etc., supplied exogenously. The final

section called 'Budgetary Calculations' accepts as input the output from the previous sections, and computes the department and faculty budgets, using many parameters supplied exogenously.

The basic objective of the CAMPUS model is to serve as a tool for educational administrators of colleges or universities in the development and analysis of alternative long range plans and annual budget. The main focus of the system is on academic program planning and the calculation of resources required to conduct these programs. The third section of this model was of considerable interest to the author. Unfortunately, the details of their work and their computer program are proprietary.

1.2.3 UNIVERSITY OF ILLINOIS.

Bareither and Schillinger (1) in their book have described the work done at the University of Illinois, in translating the educational program into physical facility requirements. They have developed a procedure called 'Numeric Method' which provides the framework for translating the educational program into physical facility requirements. Their book served as an invaluable reference during all development stages of this work. The author has borrowed many details either in full or in part from this work. In particular this book gives valuable hints in the area of efficient data management.

1.2.4 WICHE'S EFFORTS.

The Western Interstate Commission for Higher Education (3) in cooperation with the American Association of Collegiate Registrars and Admission Officers, have done invaluable work in this field. They have summarized their efforts in seven manuals. The manuals deal in great detail with the projection

procedure for different types of facilities in an educational institution. Their facilities planning and management manuals can be looked upon as 'Handbook for Facilities Planning'.

1.2.5 WORK DONE BY CRS.

CRS is a consulting firm of Architects, Planners, and Engineers. They worked for the Board of Regents, State of Kansas, and presented a facilities planning report for all six Regents' colleges and universities in the state. Their findings have been summarized in the Physical Development Manual (6). WICHE's manuals have been CRS's principal reference.

CRS has presented all the facilities planning reports, with prime emphasis on the program classification structure. This is quite understandable, since most of the recent literature in the field of educational planning gives undue empahsis to this concept. However, this concept being a new one, many people in the field have yet to become conversant with it. Hency any facilities planning report categorized only on program classification structure will be very hard to understand, possibly even undecipherable. For better understanding and evaluation of the problem at hand, it is preferable to have planning reports summarized on organizational units. The summary by program classification structure has yet another disadvantage -- the possibility of misclassification of space. It is easier to associate spaces with a particular organizational unit, than with a particular function (PCS). The summary by program classification structure is adequate only for comparing the facilities standards of the different institutions.

CRS's efforts served as one of the main references for the work undertaken and presented in this report.

1.3 PROPOSAL.

An effort will be made to provide detailed facilities planning reports for Kansas State University. A complete data processing setup will be developed for this purpose. For investigative purposes, the personnel, contact hours, existing space data, etc., for the fall semester 1972 will be used. The data processing programs will be designed to provide summary reports by department, college, university and program classification structure.

CHAPTER 2

DESCRIPTION OF THE PLANNING PROCESS

2.1 INTRODUCTION.

The translation of the educational program into physical facility requirements involves a constant evaluation and reevaluation of enrollment projections, changes in the educational program, inventory of existing facilities and development of new facilities dependant upon departmental requirements.

The manner in which a university organizes to accomplish translating the educational program into physical facility requirements is left to the university administrators. The purpose of this chapter is to give an overview of the total planning process, and then to acquaint the reader with important definitions which frequently occur.

2.2 BASIC STEPS IN THE PLANNING PROCESS.

The steps to be followed are shown in Figure 2.1. Each step must be constantly reviewed from time to time to reflect the dynamic changes taking place in a university (1, p. 3-5).

2.3 DEFINITION OF IMPORTANT TERMS.

In a work of this kind there are a few terms which occur very frequently. It is imperative that the workers in this field be familiar with their definitions. For a more detailed list, the readers are referred to WICHE (3, vol. 7).

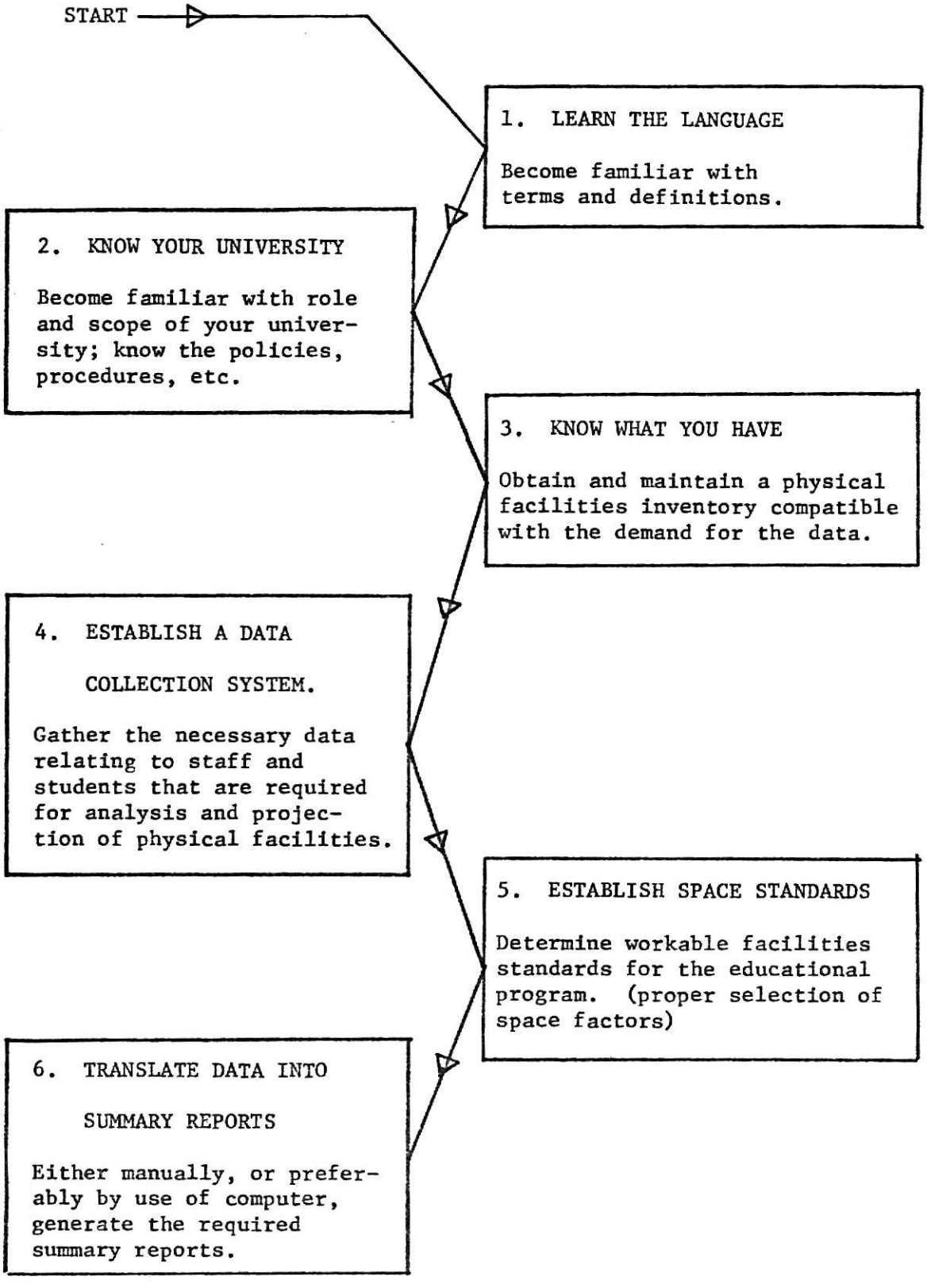


Fig. 2.1. Facilities Planning Process

Higher Educational General Information Survey (HEGIS): The annual survey of college and university statistical data conducted by the United States office of Education, the national center for educational statistics. In this survey each field of instruction was allotted a four digit identification code. Appendix D is a complete list of the HEGIS codes (3).

Administration Code or Department Code: An identification code for each department on campus. At Kansas State University there was no one unique identification code for the departments. For reasons explained in Section 4.3.3 the three digit code used by the accounting office was made into a four digit code, the first digit identifying the activity a particular department is engaged in.

Net Assignable Square Feet (NASF): The sum of all areas on all floors of a building assigned to, or available for assignment to, an occupant, including every type of space functionally usable by an occupant.

For a single room, this is the sum of all areas located between the principal surface of the walls and partitions at or near floor level. Space occupied by alcoves, closets, and built-in shelves opening into and serving the room should be included. Areas of columns, door-swings, and impaired headroom, and space occupied by heating devices may be ignored. If, however, any of these structural features constitutes a large loss of usable space, the area should be deducted from the square feet measurement of the room (3).

Program Classification Structure (PCS): A standard categorization of activities developed by the National Center for Educational Statistics (NCES) to be used as a standard format for institutional program budgets. This is a two digit code. Appendix E gives a complete list of the PCS codes and their definitions.

Weekly Contact Hour (WCH): A unit of measure which is equivalent to one hour of instruction given to one student in one week. For a particular course the cumulative WCH can be computed using the relation:

WCH = (number of hours per week the course is scheduled to meet) * (number of students enrolled in the course)

Room Utilization Rate (RUR): The number of hours per week the room is scheduled for use (3).

Station: The total facilities necessary to accomodate one person for one time period. The time period varies for different types of facilities. For example, for classroom stations, the period of time may be one hour or one class period, and for office stations, the time period may be one year (or it may be indefinite) (3).

Station Utilization Rate (SUR): The number of hours per week that a station is scheduled for use. (3)

Station Occupancy Ratio (SOR): The proportion of stations used when the room is scheduled for use. The Percent Station Utilization Rate refers to the same thing.

SOR = SUR/RUR (3).

Activity Load (Load Indicator): Signifies in quantitative terms the demand which is or would be generated by present or future activities in an institution. They may apply to staff work load, faculty occupancy load, or student course and study loads.

Full Time Equivalent (FTE): The equivalent of one person who is deemed to be carrying full load or having full time appointment, on an institutionally agreed upon convention for converting numbers of specific individuals (students or employees) to an equivalent number of full time persons (3).

Head Count (HC): Any individual considered by the institution to be a faculty member (or student) without regard to work load being performed. Includes both full time and part time members (faculty or students) (3).

Space Factor (SF): An index which indicates the amount of space in NASF per unit of projected Activity Load. Some examples follow:

a) NASF per WCH: used when projecting classroom or class laboratory space.

b) NASF per HC: when projecting research space.

c) NASF per FTE: (Faculty, staff or students) when projecting office space, study space, etc.

2.4 ROOM TYPES CLASSIFICATION AND PROJECTION TECHNIQUE.

Each separate identifiable type of space has been assigned a three digit code. Categories of rooms are in accordance with the HEGIS classification system. There are essentially thirteen classifications with each having minor subclassifications. A complete list of the room types and their definitions are given in Appendix F. For space projection purposes only the major classifications of room types were taken into consideration. A brief description of the major room types and their space projection procedures will be outlined below.

100 CLASSROOM FACILITIES:

Definition: A room used by classes which do not require special purpose equipment for student use.

Projection Method:

Activity Load: WCH

Space Factor: NASF per WCH

= (Sq. ft/Station) / (WRH*SOR)

Typical values used for Kansas State University:

Square feet per Station = 18

WRH = 30

SOR = 0.6

and hence Space Factor = 1.0 NASF/WCH.

200 CLASS LABORATORY FACILITIES:

Definition: A room used by regularly scheduled classes which require special purpose equipment for student participation, experimentation, observation, or practice in a field of study.

Projection Method:

Activity Load: WCH

Space Factor: NASF per WCH

The method for computing the space factor is the same as for the classroom space. However unlike the classroom space, the area required per WCH will vary, depending upon the field of study and level of instruction. Using the guideline values for the NASF per station, SOR and weekly room hours given in WICHE (3), the space factors were computed. They are shown in Table 2.1.

UNIT FLOOR CRITERIA : CLASS LABORATORY

HEGIS	DEPT-NAME		SQFT/ STN	RUR	SOR	SPACE-FACTOR
0108	HORTICUL & FORESTRY	L	60.00	18.00	0.75	4.44
		U	70.00	18.00	0.65	5.98
		G	80.00	16.00	0.50	10.00
0199	MILLING INDUSTRIES	L	60.00	18.00	0.75	4.44
		U	70.00	18.00	0.65	5.98
		G	80.00	16.00	0.50	10.00
0200	GENERAL ARCH	L	55.00	16.00	0.75	4.58
		U	55.00	16.00	0.75	4.58
		G	55.00	16.00	0.75	4.58
0202	ARCITECTURE	L	55.00	16.00	0.75	4.58
		U	60.00	16.00	0.65	5.77
		G	70.00	16.00	0.50	8.75
0203	INTERIOR ARCH	L	70.00	16.00	0.65	6.73
		U	70.00	16.00	0.65	6.73
		G	70.00	16.00	0.65	6.73
0204	LANDSCAPE ARCH	L	70.00	16.00	0.65	6.73
		U	70.00	16.00	0.65	6.73
		G	70.00	16.00	0.65	6.73
0206	REG & COMM PLANNING	L	65.00	16.00	0.65	6.25
		U	65.00	16.00	0.65	6.25
		G	65.00	16.00	0.65	6.25
0299	CONSTRUCTION SCIENCE	L	55.00	16.00	0.75	4.58
		U	60.00	16.00	0.65	5.77
		G	70.00	16.00	0.50	8.75
0401	BIOLOGY	L	50.00	18.00	0.75	3.70
		U	60.00	18.00	0.65	5.13
		G	70.00	16.00	0.50	8.75
0404	PLANT PATHOLOGY	L	55.00	18.00	0.75	4.07
		U	60.00	18.00	0.65	5.13
		G	70.00	16.00	0.50	8.75
0414	BIO CHEMISTRY	L	60.00	18.00	0.75	4.44
		U	70.00	18.00	0.65	5.98
		G	80.00	16.00	0.50	10.00
0421	ENTOMOLOGY	L	60.00	18.00	0.75	4.44
		U	70.00	18.00	0.65	5.98
		G	80.00	16.00	0.50	10.00
0506	BUSINESS ADMN	L	35.00	20.00	0.65	2.69
		U	35.00	20.00	0.65	2.69
		G	35.00	20.00	0.65	2.69

Table 2.1. (continued)

UNIT FLOOR CRITERIA : CLASS LABORATORY

HEGIS	DEPT-NAME		SQFT/ STN	RUR	SOR	SPACE-FACTOR
0602	TECHNICAL JOURNALISM	L	45.00	24.00	0.70	2.68
		U	45.00	24.00	0.70	2.68
		G	45.00	24.00	0.70	2.68
0701	COMPUTER SCIENCE	L	70.00	22.00	0.70	4.55
		U	70.00	22.00	0.70	4.55
		G	70.00	22.00	0.70	4.55
0807	ADULT & OCCUP	L	35.00	20.00	0.75	2.33
		U	35.00	20.00	0.75	2.33
		G	35.00	20.00	0.75	2.33
0827	ADMN & FOUNDATIONS	L	35.00	20.00	0.75	2.33
		U	35.00	20.00	0.75	2.33
		G	35.00	20.00	0.75	2.33
0829	CURR & INSTR	L	35.00	20.00	0.75	2.33
		U	35.00	20.00	0.75	2.33
		G	35.00	20.00	0.75	2.33
0900	GENERAL ENGG	L	110.00	20.00	0.70	7.86
		U	110.00	20.00	0.70	7.86
		G	110.00	20.00	0.70	7.86
0902	AEROSPACE	L	65.00	16.00	0.65	6.25
		U	65.00	16.00	0.65	6.25
		G	65.00	16.00	0.65	6.25
0903	AGRICULTURAL ENGG	L	120.00	20.00	0.65	9.23
		U	120.00	20.00	0.65	9.23
		G	120.00	20.00	0.65	9.23
0906	CHEMICAL ENGG	L	170.00	20.00	0.65	13.08
		U	170.00	20.00	0.65	13.08
		G	170.00	20.00	0.65	13.08
0908	CIVIL ENGG	L	170.00	20.00	0.65	13.08
		U	170.00	20.00	0.65	13.08
		G	170.00	20.00	0.65	13.08
0909	ELECTRICAL ENGG	L	100.00	20.00	0.65	7.69
		U	100.00	20.00	0.65	7.69
		G	100.00	20.00	0.65	7.69
0910	MECHANICAL ENGG	L	170.00	20.00	0.65	13.08
		U	170.00	20.00	0.65	13.08
		G	170.00	20.00	0.65	13.08
0913	INDUSTRIAL ENGG	L	140.00	20.00	0.65	10.77
		U	140.00	20.00	0.65	10.77
		G	140.00	20.00	0.65	10.77
0920	NUCLEAR ENGG	L	170.00	20.00	0.65	13.08

UNIT FLOOR CRITERIA : CLASS LABORATORY

HEGIS	DEPT-NAME		SQFT/ STN	RUR	SOR	SPACE-FACTOR
		U	170.00	20.00	0.65	13.08
		G	170.00	20.00	0.65	13.08
0999	APPLIED MECHANICS	L	110.00	20.00	0.70	7.86
		U	110.00	20.00	0.70	7.86
		G	110.00	20.00	0.70	7.86
1002	ART	L	55.00	18.00	0.65	4.70
		U	55.00	18.00	0.65	4.70
		G	55.00	18.00	0.65	4.70
1005	MUSIC	L	25.00	22.00	0.70	1.62
		U	25.00	22.00	0.70	1.62
		G	25.00	22.00	0.70	1.62
1101	MODERN LANGUAGES	L	45.00	24.00	0.70	2.68
		U	45.00	24.00	0.70	2.68
		G	45.00	24.00	0.70	2.68
1218	VETERINARY MEDICINE	L	180.00	20.00	0.65	13.85
		U	180.00	20.00	0.65	13.85
		G	180.00	20.00	0.65	13.85
1299	INFECTIOUS DISEASES	L	180.00	20.00	0.65	13.85
		U	180.00	20.00	0.65	13.85
		G	180.00	20.00	0.65	13.85
1299	PATHOLOGY	L	180.00	20.00	0.65	13.85
		U	180.00	20.00	0.65	13.85
		G	180.00	20.00	0.65	13.85
1299	PHYSIOLOGICAL SC	L	180.00	20.00	0.65	13.85
		U	180.00	20.00	0.65	13.85
		G	180.00	20.00	0.65	13.85
1299	SURGERY & MEDICINE	L	180.00	20.00	0.65	13.85
		U	180.00	20.00	0.65	13.85
		G	180.00	20.00	0.65	13.85
1300	DEAN OF HOME ECON	L	65.00	20.00	0.65	5.00
		U	65.00	20.00	0.65	5.00
		G	65.00	20.00	0.65	5.00
1303	CLOTHING & TEXTILES	L	60.00	20.00	0.65	4.62
		U	60.00	20.00	0.65	4.62
		G	60.00	20.00	0.65	4.62
1304	FAMILY ECONOMICS	L	50.00	20.00	0.65	3.85
		U	50.00	20.00	0.65	3.85
		G	50.00	20.00	0.65	3.85
1305	FAMILY & CHILD DEV	L	50.00	20.00	0.65	3.85
		U	50.00	20.00	0.65	3.85

Table 2.1. (continued)

UNIT FLOOR CRITERIA : CLASS LABORATORY

HEGIS	DEPT-NAME		SQFT/ STN	RUR	SOR	SPACE-FACTOR
		G	50.00	20.00	0.65	3.85
1306	FOODS & NUTRITION	L	70.00	20.00	0.65	5.38
		U	70.00	20.00	0.65	5.38
		G	70.00	20.00	0.65	5.38
1307	INST MANAGEMENT	L	70.00	20.00	0.65	5.38
		U	70.00	20.00	0.65	5.38
		G	70.00	20.00	0.65	5.38
1501	ENGLISH	L	30.00	24.00	0.70	1.79
		U	30.00	24.00	0.70	1.79
		G	30.00	24.00	0.70	1.79
1501	MILITARY SCIENCE	L	45.00	20.00	0.70	3.21
		U	45.00	20.00	0.70	3.21
		G	45.00	20.00	0.70	3.21
1506	SPEECH	L	85.00	20.00	0.65	6.54
		U	85.00	20.00	0.65	6.54
		G	85.00	20.00	0.65	6.54
1509	PHILOSOPHY	L	30.00	24.00	0.75	1.67
		U	30.00	24.00	0.75	1.67
		G	30.00	24.00	0.75	1.67
1702	STATISTICS	L	30.00	24.00	0.75	1.67
		U	30.00	24.00	0.75	1.67
		G	30.00	24.00	0.75	1.67
1902	PHYSICS	L	55.00	20.00	0.75	3.67
		U	60.00	18.00	0.65	5.13
		G	80.00	16.00	0.50	10.00
1905	CHEMISTRY	L	50.00	18.00	0.75	3.70
		U	50.00	18.00	0.75	3.70
		G	50.00	18.00	0.75	3.70
1914	GEOLOGY	L	55.00	20.00	0.70	3.93
		U	60.00	18.00	0.65	5.13
		G	70.00	18.00	0.55	7.07
2001	PSYCHOLOGY	L	50.00	20.00	0.75	3.33
		U	65.00	18.00	0.65	5.56
		G	75.00	16.00	0.50	9.38
2205	MATHEMATICS	L	30.00	24.00	0.70	1.79
		U	30.00	24.00	0.70	1.79
		G	30.00	24.00	0.70	1.79
2206	GEOGRAPHY	L	55.00	20.00	0.70	3.93
		U	60.00	18.00	0.65	5.13
		G	70.00	18.00	0.55	7.07

Table 2.1. (continued)

UNIT FLOOR CRITERIA : CLASS LABORATORY

HEGIS	DEPT-NAME		SQFT/ STN	RUR	SOR	SPACE-FACTOR
2206	HISTORY	L	30.00	24.00	0.70	1.79
		U	30.00	24.00	0.70	1.79
		G	30.00	24.00	0.70	1.79
2207	POLITICAL SCIENCE	L	30.00	22.00	0.70	1.95
		U	30.00	22.00	0.70	1.95
		G	30.00	22.00	0.70	1.95
2208	SOC & ANTHROPOLOGY	L	30.00	24.00	0.75	1.67
		U	30.00	24.00	0.75	1.67
		G	30.00	24.00	0.75	1.67

250 NON CLASS LABORATORY (RESEARCH):

Definition: A room used for laboratory applications, research and/or training in research methodology, which require special purpose equipment for staff and/or student experimentation of observation.

Projection Method:

Activity Load: HC

Space Factor: NASF per HC

The space factors were the guideline values given in WICHE (3), and are shown in Table 2.2. Bareither (1) has outlined a more elaborate procedure for computing these space factors, as follows:

Space Factor = (RDU*RDF) NASF/UNIT

where,

RDU = Research Demand Units/UNIT,

RDF = Research Demand Factor

UNIT = FTE or HC

"The research demand unit" concept is based on the premise that the research space is generated by the FTE research faculty, FTE teaching faculty, and HC graduate students classified by level as beginning or advanced graduates.

It is recognized that the FTE research faculty require more research space than the FTE teaching faculty. The FTE research faculty devote 100 percent of their time to research, whereas the FTE teaching faculty are expected to devote approximately 20 percent of their time to research. In addition, an advanced graduate student devotes more of his time to individual research than does the beginning graduate student, whose time is devoted mainly to scheduled classes, seminars, and library research. To account

ACADEMIC PROGRAM	ASSIGNABLE Sq. Ft. PER FACULTY MEMBER *	HEAD COUNT GRADUATE STUDENTS ACCOMODATED IN THE NASF PROVIDED FOR EACH FACULTY MEMBER	ADDITIONAL NASF PER ADDITIONAL GRADUATE STUDENT ENGAGED IN RESEARCH
Agriculture & Natural Resources Engineering Biological Sciences Physical Sciences	900-1300	4	200-250
Architech & Env Design Fine and Applied Arts Home Economics Psychology Communications	600-900	4	150-200
Education Area Studies Business and Management Computer & Info Sciences Foreign Languages Letters Library Science Mathematics Public Affairs & Services Law Theology	150-200	4	20-25

*Includes service space.

Table 2.2. General Planning Criteria for Research Space (WICHE)

for these variations in space requirements, is devoted mainly to scheduled classes, seminars, and library research. To account for these variations in space requirements, the RDU are different for each of the above four cases. To account for the variations in space requirement of a given field of study, a concept called the 'research demand factor' has been developed. Table 2.3 gives the value of RDU, RDF and space factors computed for each field of study at the Kansas State University. These values were not used in this report, principally because the graduate student data is not available by the two levels (beginning and advanced). However the computer program has been designed to accept these data, when they become available, and make the projection.

300 OFFICE:

Definition: A room used by faculty, staff, or students working at a desk (or table).

Projection Method:

Activity Load: Either FTE or HC data depending on institutional policy for allotting office space. For the Kansas State University, the FTE data was used.

Space Factor: Either NASF per FTE or NASF per HC. Since this class of space also takes into account the conference rooms, the space factor chosen must take care of the allowance for these.

400 STUDY FACILITIES:

Definition: A room used to study books or audio visual materials on an individual basis.

Projection Method: The study room space is dependant upon the total student enrollment. In many cases the standards have been 30 square feet per station with a station capacity of 25 percent of the student body (1). These

Table 2.3.

UNIT FLOOR CRITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
0203	INTERIOR ARCH	1	15	20.0	300.00
		2	3	20.0	60.00
		3	3	20.0	60.00
		4	15	20.0	300.00
0204	LANDSCAPE ARCH	1	15	20.0	300.00
		2	3	20.0	60.00
		3	3	20.0	60.00
		4	15	20.0	300.00
0206	REG & COMM PLANNING	1	15	20.0	300.00
		2	3	20.0	60.00
		3	3	20.0	60.00
		4	15	20.0	300.00
0299	CONSTRUCTION SCIENCE	1	15	20.0	300.00
		2	3	20.0	60.00
		3	3	20.0	60.00
		4	15	20.0	300.00
0401	BIOLOGY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0404	PLANT PATHOLOGY	1	15	26.0	390.00
		2	3	26.0	78.00
		3	3	26.0	78.00
		4	15	26.0	390.00
0414	BIO CHEMISTRY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0421	ENTOMOLOGY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0506	BUSINESS ADMN	1	15	1.0	15.00
		2	3	1.0	3.00
		3	3	1.0	3.00
		4	15	1.0	15.00
0602	TECHNICAL JOURNALISM	1	15	1.0	15.00

Table 2.3. (continued)

UNIT FLOOR CRITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
		2	3	1.0	3.00
		3	3	1.0	3.00
		4	15	1.0	15.00
0701	COMPUTER SCIENCE	1	15	10.0	150.00
		2	3	10.0	30.00
		3	3	10.0	30.00
		4	15	10.0	150.00
0829	CURR & INSTR	1	15	5.0	75.00
		2	3	5.0	15.00
		3	3	5.0	15.00
		4	15	5.0	75.00
0835	PHYSICAL EDUCATION	1	15	8.0	120.00
		2	3	8.0	24.00
		3	3	8.0	24.00
		4	15	8.0	120.00
0903	AGRICULTURAL ENGG	1	15	45.0	675.00
		2	3	45.0	135.00
		3	3	45.0	135.00
		4	15	45.0	675.00
0906	CHEMICAL ENGG	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
0908	CIVIL ENGG	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
0909	ELETRICAL ENGG	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0910	MECHANICAL ENGG	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0913	INDUSTRIAL ENGG	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00

UNIT FLOOR CRITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
		4	15	25.0	375.00
0920	NUCLEAR ENGG	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
0999	APPLIED MECHANICS	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1002	ART	1	15	35.0	525.00
		2	3	35.0	105.00
		3	3	35.0	105.00
		4	15	35.0	525.00
1005	MUSIC	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
1101	MODERN LANGUAGE	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
1213	VETERINARY MEDICINE	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
1299	INFECTIOUS DISEASES	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
1299	PATHOLOGY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
1299	PHYSIOLOGICAL SC	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00

UNIT FLOOR CPITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
1299	SURGERY & MEDICINE	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1300	DEAN OF HOME ECON	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1300	INDEP STUDY LAB	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1303	CLOTHING & TEXTILES	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1304	FAMILY ECON	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1305	FAMILY & CHILD DEV	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1306	FOODS & NUTRITION	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1307	INST MANAGEMENT	1	15	30.0	450.00
		2	3	30.0	90.00
		3	3	30.0	90.00
		4	15	30.0	450.00
1501	ENGLISH	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
1501	MILITARY SCIENCE	1	15	0.5	7.50
		2	3	0.5	1.50

Table 2.3. (continued)

UNIT FLOOR CRITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
		3	3	0.5	1.50
		4	15	0.5	7.50
1506	SPEECH	1	15	8.3	124.45
		2	3	8.3	24.89
		3	3	8.3	24.89
		4	15	8.3	124.45
1509	PHILOSOPHY	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
1702	STATISTICS	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
1902	PHYSICS	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
1905	CHEMISTRY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
1914	GEOLOGY	1	15	25.0	375.00
		2	3	25.0	75.00
		3	3	25.0	75.00
		4	15	25.0	375.00
2001	PSYCHOLOGY	1	15	15.0	225.00
		2	3	15.0	45.00
		3	3	15.0	45.00
		4	15	15.0	225.00
2205	MATHEMATICS	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
2206	GEOGRAPHY	1	15	5.0	75.00
		2	3	5.0	15.00
		3	3	5.0	15.00
		4	15	5.0	75.00

Table 2.3 (continued)

UNIT FLOOR CRITERIA : NON CLASS LAB

1 FTE RESEARCH FACULTY
2 FTE TEACHING FACULTY
3 HEAD COUNT G1
4 HEAD COUNT G2

HEGIS	DEPATRMENT		RDU	RDF	SPACE-FACTOR
2206	HISTORY	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50
2207	POLITICAL SCIENCE	1	15	0.5	7.50
		2	3	0.5	1.50
		3	3	0.5	1.50
		4	15	0.5	7.50

standards are believed to be adequate for undergraduate reading room space; but as the graduate program of an institution (or department) increases, there is a need for carrel space for both graduate students and faculty. The demand for carrel space varies by the field of study. Those fields of study that do not require laboratory research space will require more carrel space for both graduate students and faculty. Based on these premises, the following standards are recommended for reading room space (1):

1. 7.5 square feet per FTE undergraduate student.
2. 7.5 square feet per HC graduate student.
3. 7.5 square feet per advanced graduate student in those fields with high research requirement. (fields of study that have a RDF greater than one).
4. 15 square feet per HC advanced graduate student in those fields of study with low research requirements (RDF of one or less).
5. 15 square feet per FTE teaching and research faculty in those departments that have low research requirements (those departments with a RDF of one or less).
6. 3 square feet per FTE teaching and research faculty in those departments that have high research requirements (RDF of greater than one).

However the above suggestion could not be implemented because some of the needed data, (FTE undergraduate students and headcount graduate students by level of beginning and advanced) is available only for different curricula, and not for the different departments. There is no one-to-one relationship between curricula and departments. Hence this seemed to be a poor method of projection, since the data is not readily available. The projected space was set equal to the existing space for this room type.

However the computer program has been designed to accept the needed data, when it becomes available, and make projections.

420 STACK SPACE:

Definition: A room (or portion of a room) used to provide shelving for books or audio visual materials, used by staff and/or students on an individual basis.

Projection Method: (Ref. 6)

Data Required: The sum of the number of bound volumes, the number of printed government documents that were not reported as bound volumes, and the number of reels of microfilm.

Space Factor: NASF per unit of the above data.

440 LIBRARY PROCESSING ROOM:

Definition: A room which serves a study room or a stack room.

Projection Method: Usually a percentage of the sum areas of room types 410 and 420 is allotted for this purpose (6).

500 SPECIAL USE FACILITIES:

This category includes Armory Facilities, Athletic Facilities, Audio-Visual Service, Clinic Facilities, Demonstration Facilities, Field Service Facilities, etc.

Projection Method: None developed.

600 GENERAL USE FACILITIES:

This category includes another set of widely varying types of space. These are Assembly Facilities, Exhibition Facilities, Food Facilities, Health Facilities, Lounge Facilities, Merchandising Facilities, Recreation Facilities, etc.

Projection Method: None developed.

700 SUPPORTING FACILITIES:

This category includes another set of widely varying types of space. These are Data Processing Computer Facilities, Shop Facilities, Storage Facilities, Vehicle Storage, Central Food Service, Central Laundry, etc.

Projection Method: None developed.

Brief Discussion on Room Types 500, 600, and 700:

As seen, under these three categories there is a large number of different types of space. No single one of these space types can be related firmly to a readily measurable variable within the institution. Rather, the amount of these space types available or required by an institution is determined by the institution's philosophies, organizational structure, operating style, governing board policies, and financial capabilities. In addition there are certain substitution effects evident among these space categories. For example lounge and recreation space may be reduced in order to acquire more athletic facilities, or assembly facilities (3, Vol. 6, p. 90).

WICHE (3, Vol. 6) recommends about 40 to 65 percent of the total area of room types 100 to 440, be allotted for this category. However, this suggestion has not been implemented in this work. The projected space has been set equal to the existing space.

800 MEDICAL CARE FACILITIES:

This category includes varying types of medical service facilities. Some of them are Human Hospital Clinic Facilities, Dental Clinic Service, Veterinary Hospital Clinic Facilities, etc.

Projection Method: A study of this type of space was not made. Bareither (1) has given valuable hints. However these suggestions were not implemented. The projected space has been set equal to the existing space.

900 RESIDENTIAL FACILITIES:

This space category includes residence for single persons, married persons, dormitory, food service in residence halls, etc.

Projection Method: A study of this space category was not made. The procedure given in CRS (6) has been implemented. However CRS does not have any discussion on the method employed. The data that have been used are the head count for single persons, head count for married persons, and a suitable space factor (NASF per HC).

CHAPTER 3

DETAILS OF THE NEW SYSTEM

3.1 INTRODUCTION.

This chapter will outline the new facilities planning system developed for Kansas State University. However, it should be noted that this system is not a purely original one. It is only an attempt to refine and restructure the system already existing, the one developed by CRS. The description presented here will also serve to explain the main theme of the work done by CRS.

3.2 PLANNING REPORTS.

3.2.1 GENERAL.

Within any educational institution, there are essentially two principal modes of categorization of efforts. In one of these modes the individual departments are the basic organizational units; aggregation of departments with similar types of effort is the next higher unit, namely the college. The other mode of categorization is by the Program Classification Structure (Appendix E). However, it is apparent, that these modes are interdependant; they are essentially subsets of each other. Any Facilities Planning System can present summary reports with prime importance given to one of these modes. In the reports presented by CRS, PCS was chosen as the principal mode of categorization. However the concept of PCS is relatively new, and many people in the planning field have yet to become familiar with it. Hence the summary reports presented by CRS are difficult to digest. These reports

are found adequate for comparing the facilities standard of the six schools in the state, but inadequate for effective space planning within a particular school. It is easier for the administrators to seek a solution to the problem of effective space allocation and utilization, when the planning reports summarize space by the different organizational units. It is easier to identify the space surplus/shortage, and plan on reallocation, or on building new facilities. Hence in this effort, facilities planning reports will be presented for the different organizational units existing in a university.

3.2.2 DEPARTMENT SPACE SUMMARY.

Within a particular department, the space summary will be sequenced on room types, and PCS within room types. Table 3.1 is a sample of this summary report. As seen, some room types have subcategories. In these cases the projected NASF is given by the subcategories present; a subtotal of these projected spaces is compared with the existing space for this room type and the associated PCS. If a room type has more than one PCS associated with it, there is a total given which serves to compare the net projected and existing space for this room type. The projected space has been set equal to the existing space, in cases where a proper function (need) cannot be associated for that space type (perhaps due to error in existing space classification), or in cases where no projection method exists. Such cases have been flagged with a '##' sign. The grand totals printed are the overall totals of the projected and existing space of all the space types, existing in this department. Since, at Kansas State University, the classrooms are a common property of all the departments, the projected NASF for classrooms

Table 3.1.

KANSAS STATE UNIVERSITY : DEPARTMENT SPACE SUMMARY

HEGIS	DEPT CODE	DESCRIPTION	ROOM-TYPE		PCS ACTIVITY LOAD	UNIT	SPACE-FACTOR	PROJECTED NASF	EXISTING NASF
0100	4200	VP FOR AGRICULTURE							
		CLASS LABORATORY	200	## 11				156	156
		NON CLASS LAB (RES)	250	FAC 21	5.00	HC	1300.00	4500	
				STF 21	10.00	HC	250.00	2500	
				SUB-TOT 21				9000	394
				FAC 22	7.00	HC	1300.00	9100	
				STF 22	5.00	HC	250.00	1250	
				SUB-TOT 22				10350	00
				TOT				19350	394
		OFFICE SPACE	300	FAC 11	0.70	FTE	170.00	119	
				GR 11	0.00	HC	60.00	00	
				STF 11	0.50	FTE	170.00	85	
				SUB-TOT 11				204	00
				FAC 21	3.10	FTE	170.00	527	
				STF 21	10.00	FTE	170.00	1700	
				SUB-TOT 21				2227	1320
				FAC 22	6.00	FTE	170.00	1020	
				GR 22	0.00	FTE	170.00	00	
				STF 22	4.50	FTE	170.00	765	
				SUB-TOT 22				1785	00
				STF 33	0.30	FTE	170.00	51	00
				## 46				1938	1938
				TOT				6205	3258
		GRAND-TOTAL						25711	3809
0100	4201	DEAN OF AGRICULTURE							
		CLASS ROOM	100		11 508.00	WCH	1.00	508	00
		CLASS LABORATORY	200	## 11				60	60
		NON CLASS LAB (RES)	250	FAC 22	2.00	HC	1300.00	2600	00
		OFFICE SPACE	300	FAC 11	3.90	FTE	170.00	663	
				GR 11	0.00	FTE	60.00	00	
				STF 11	2.00	FTE	170.00	340	
				SUB-TOT 11				1003	75
				FAC 22	1.10	FTE	170.00	187	
				GR 22	0.00	FTE	170.00	00	
				SUB-TOT 22				187	00
				## 46				907	907
				TOT				2097	982
		STUDY FACILITIES	410	## 46				1040	1040
		GRAND-TOTAL						5797	2082
0111	1202	AGRICUL ECONOMICS							

Table 3.1. (continued)

KANSAS STATE UNIVERSITY : DEPARTMENT SPACE SUMMARY

HEGIS	DEPT CODE	DESCRIPTION	ROOM-TYPE		PCS	ACTIVITY LOAD	UNIT	SPACE-FACTOR	PROJECTED NASF	EXISTING NASF
				STF	11	4.00	FTE	170.00	680	
				SUB-TOT	11				3725	4249
				FAC	22	2.70	FTE	170.00	459	
				GR	22	0.00	FTE	170.00	00	
				STF	22	0.00	FTE	170.00	00	
				SUB-TOT	22				459	433
				TOT					4184	4678
		STUDY FACILITIES	410	##	11				374	374
		SUPPORTING FAC	700	##	11				631	631
		GRAND-TOTAL							17378	22354
0913	1409	INDUSTRIAL ENGG								
		CLASS ROOM	100		11	647.00	WCH	1.00	647	00
		CLASS LABORATORY	200	FS	11	567.00	WCH	10.80	6123	
				JS	11	63.00	WCH	10.80	680	
				G1	11	0.00	WCH	10.80	00	
				G2	11	24.00	WCH	10.80	259	
				SUB-TOT	11				7062	8690
				##	22				70	70
				TOT					7132	8760
		NON CLASS LAB (RES)	250	##	11				192	192
				GR	22	3.00	HC	250.00	750	
				FAC	22	5.00	HC	1300.00	6500	
				STF	22	3.00	HC	250.00	750	
				SUB-TOT	22				8000	332
				TOT					8192	1024
		OFFICE SPACE	300	FAC	11	12.90	FTE	170.00	2193	
				GR	11	3.00	HC	60.00	180	
				STF	11	3.00	FTE	170.00	510	
				SUB-TOT	11				2883	5744
				FAC	22	2.30	FTE	170.00	391	
				GR	22	0.00	FTE	170.00	00	
				STF	22	3.00	FTE	170.00	510	
				SUB-TOT	22				901	250
				TOT					3784	5994
		STUDY FACILITIES	410	##	11				900	900
		GENERAL USE FAC	600	##	11				123	123
		SUPPORTING FAC	700	##	71				10544	10544
				##	11				2397	2397
				TOT					12941	12941
		GRAND-TOTAL							33072	29742
0910	1410	MECHANICAL ENGG								

does not enter the respective grand total. The existing NASF shown is the classroom service areas which belong to that particular department.

3.2.3 COLLEGE SPACE SUMMARY.

Two space summary reports have been presented for each college. College Space Summary I, samples of which are shown in Table 3.2, is identical to the department space summary report in format. College Space Summary II (Table 3.3) summarizes the surplus/deficit of space for each room type. The surplus denotes the excess of existing NASF over projected NASF. The entry under '%' is the percentage surplus/deficit (expressed as a percentage of the existing NASF). In cases where existing space is zero, an asterisk has been printed. In cases where the projected NASF is equal to the existing space, the entries are zero. For reasons explained in Section 3.2.2, the classroom space has not been shown in this report.

3.2.4 UNIVERSITY SPACE SUMMARY.

Table 3.4 shows some samples from this summary report. The interpretation is similar to that of department space summary report.

3.2.5 SUMMARY BY PROGRAM CLASSIFICATION STRUCTURE.

This report (Table 3.5) summarizes the projected space, existing space, and surplus or deficit, by the different program classification structures (all minor breakdowns) existing in the university. The surplus denotes the excess of existing NASF over projected NASF, and has also been expressed as a percentage of the existing NASF. In cases where the existing NASF is zero, an asterisk has been printed under the percentage entry. For each PCS major category, a subtotal (of relevant quantities) of all its subcategories has

Table 3.2.

KANSAS STATE UNIVERSITY : COLLEGE SPACE SUMMARY I

ARCHITECH & DESIGN HEGIS: 0200

DESCRIPTION	ROOM-TYPE		PCS	ACTIVITY LOAD	UNIT	PROJECTED NASF	EXISTING NASF
CLASS ROOM	100		11	6027.0	WCH	6027	00
CLASS LABORATORY	200	FS	11	3850.0	WCH	17848	
		JS	11	5142.0	WCH	30666	
		G1	11	1668.0	WCH	13921	
		G2	11	54.0	WCH	475	
		##	11	0.0		1423	
	SUB-TOT		11			64333	28258
		##	22	0.0		1130	1130
		TOT				65463	29388
NON CLASS LAB (RES)	250	##	11	0.0		340	340
OFFICE SPACE	300	FAC	11	45.4	FTE	7718	
		STF	11	3.0	FTE	510	
	SUB-TOT		11			8228	9214
		##	22	0.0		325	325
		STF	46	4.0	FTE	680	00
		TOT				9233	9539
STACK SPACE	420	##	11	0.0		233	233
		##	46	0.0		3019	3019
		TOT				3252	3252
GENERAL USE FAC	600	##	11	0.0		193	193
SUPPORTING FAC	700	##	11	0.0		2425	2425
GRAND-TOTAL						80906	45137

COLLEGE OF ENGG HEGIS: 0900

DESCRIPTION	ROOM-TYPE		PCS	ACTIVITY LOAD	UNIT	PROJECTED NASF	EXISTING NASF
CLASS ROOM	100		11	7715.0	WCH	7715	135
CLASS LABORATORY	200	FS	11	2706.0	WCH	29190	
		JS	11	2023.0	WCH	22931	
		G1	11	420.0	WCH	4670	
		G2	11	24.0	WCH	259	
	SUB-TOT		11			57050	59929
		##	22	0.0		9031	9031
		TOT				66081	68960
NON CLASS LAB (RES)	250	##	11	0.0		15030	15030
		GR	22	36.0	HC	9000	
		FAC	22	56.0	HC	72800	
		STF	22	23.0	HC	5750	
	SUB-TOT		22			87550	39247
		##	35	0.0		3021	3021
		TOT				105601	57298
OFFICE SPACE	300	FAC	11	89.4	FTE	15198	
		STF	11	23.5	FTE	3995	
		GR	11	36.0	HC	2160	
	SUB-TOT		11			21353	34168
		FAC	22	37.2	FTE	6276	
		STF	22	18.1	FTE	3077	
		##	22	0.0		749	
	SUB-TOT		22			10102	5163
		FAC	35	10.2	FTE	1734	
		STF	35	3.0	FTE	510	
		##	35	0.0		568	
	SUB-TOT		35			2812	3448
		STF	46	9.0	FTE	1530	903
		##	63	0.0		248	248
		TOT				36045	43930
STUDY FACILITIES	410	##	11	0.0		1788	1788
STACK SPACE	420	##	22	0.0		133	133
SPECIAL USE FAC	500	##	11	0.0		512	512
GENERAL USE FAC	600	##	11	0.0		123	123
SUPPORTING FAC	700	##	11	0.0		5683	5683
		##	22	0.0		778	778
		##	46	0.0		535	535
		##	71	0.0		10544	10544
		TOT				17540	17540

Table 3.2. (continued)

COLLEGE OF ENGG HEGIS: 0900

DESCRIPTION	ROOM-TYPE	PCS	ACTIVITY UNIT LOAD	PROJECTED NASF	EXISTING NASF
UNASSIGNABLE, ETC	688 ##	81	0.0	42	42
GRAND-TOTAL				227865	190461

Table 3.2 (continued)

KANSAS STATE UNIVERSITY : COLLEGE SPACE SUMMARY II

NOTE : THE DEF / SUR NASF IS IN HUNDREDS OF SQUARE FEET
THE % AGE SURPLUS (OR DEF) HAS BEEN EXPRESSED AS A PERCENTAGE OF THE EXISTING NASF

REGIS ADMN-CODE	COLLEGE	200 SUR-	%	250 SUR-	%	300 SUR-	%	400 SUR-	%	500 SUR-	%	600 SUR-	%	700 SUR-	%	800-900 SUR-	%
0100	AGRICULTURE COLLEGE	-153	-27	-1778	-99	-460	-67	0	0	0	*	0	0	0	0	0	0
0200	ARCHITECH & DESIGN	-360	-122	0	0	3	3	0	0	0	*	0	0	0	0	0	*
	ARTS & SCIENCES	-438	-31	392	32	-67	-6	0	0	0	0	0	0	0	0	0	*
0800	COLLEGE OF EDUCATION	-4	-8	-2	-43	-47	-55	0	*	0	*	0	*	0	*	0	*
0900	COLLEGE OF ENGG	28	4	-483	-84	78	17	0	0	0	0	0	0	0	0	0	*
1300	HOME ECONOMICS	-100	-40	-113	-93	-38	-33	0	0	0	0	0	0	0	*	0	*
1200	VETERINARY MEDICINE	-829	-795	-75	-13	-85	-38	0	0	0	*	0	*	0	0	0	0
0500	BUSINESS ADMN	17	96	0	*	-8	-13	0	0	0	*	0	*	0	*	0	*
	RES INST & CENTERS	0	*	-55	*	9	29	0	*	0	*	0	*	0	0	0	*
	PUBLIC SERVICE	0	*	0	*	-200	-71	0	0	0	0	0	0	0	0	0	*
	ACADEMIC SUPPORT	0	*	0	*	-96	-56	-432	-28	0	0	0	0	0	0	0	*
	STUDENT SUPPORT	0	*	0	*	-588	-104	0	0	0	0	0	0	0	0	0	0
	INSTITUTIONAL SUPPOR	0	*	-10	*	-456	-112	0	*	0	0	0	0	0	0	0	0
	CLASSROOMS	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0	*
	MISCELLANEOUS	0	*	0	*	0	*	0	*	0	*	0	*	0	*	0	*

Table 3.3.

Table 3.4

KANSAS STATE UNIVERSITY : SPACE SUMMARY BY ROOM TYPES.

DESCRIPTION	ROOM-TYPE	PCS	ACTIVITY LOAD	UNIT	PROJECTED NASF	EXISTING NASF	SURPLUS/ DEFICIT	% OF EXIS
CLASS ROOM	100	11	195937.0	WCH	195937			
	##	11	0.0		244			
	SUB-TOT	11			196181	122044		
	##	22	0.0		1055	1055		
	TOT				197236	123099	-74137	-60
CLASS LABORATORY	200 FS	11	48640.5	WCH	235297			
	JS	11	14196.0	WCH	42968			
	G1	11	12645.0	WCH	131609			
	G2	11	2108.0	WCH	17324			
	##	11	0.0		11019			
	SUB-TOT	11			488217	364085		
	##	22	0.0		26915	26915		
	##	36	0.0		3829	3829		
	##	71	0.0		8	8		
	TOT				518969	334837	-184132	-54
NON CLASS LAB (RES)	250 ##	11	0.0		53817	53817		
	FAC	21	5.0	HC	6500			
	STF	21	32.0	HC	8000			
	SUB-TOT	21			14500	394		
	GR	22	304.0	HC	74300			
	FAC	22	360.0	HC	439500			
	STF	22	267.3	HC	67107			
	SUB-TOT	22			561907	363493		
	##	35	0.0		3021	3021		
	##	36	0.0		6090	6090		
	##	63	0.0		263	263		
	TOT				639598	427078	-212520	-49
OFFICE SPACE	300 FAC	11	869.0	FTE	147730			
	STF	11	188.9	FTE	32113			
	GR	11	304.0	HC	18240			
	##	11	0.0		3959			
	SUB-TOT	11			202042	183413		
	FAC	21	3.1	FTE	527			
	STF	21	18.8	FTE	3196			
	SUB-TOT	21			3723	3200		
	FAC	22	300.8	FTE	51088			
	GR	22	1.9	FTE	323			
	STF	22	198.5	FTE	33745			
	##	22	0.0		2375			
	SUB-TOT	22			87531	71388		
	FAC	31	25.0	FTE	4250			
	STF	31	30.0	FTE	5100			
	SUB-TOT	31			9350	3756		
	##	32	0.0		1138	1138		
	STF	33	0.3	FTE	51	0		
	FAC	35	157.5	FTE	26775			
	STF	35	67.9	FTE	11543			
	##	35	0.0		1900			

Table 3.4. (continued)

KANSAS STATE UNIVERSITY : SPACE SUMMARY BY ROOM TYPES.

DESCRIPTION	ROOM-TYPE		PCS	ACTIVITY LOAD	UNIT	PROJECTED NASF	EXISTING NASF	SURPLUS/ DEFICIT	% OF EXIS
	SUB-TOT		35			40218	26532		
	FAC		36	54.5	FTE	9265			
	STF		36	77.7	FTE	13209			
	##		36	0.0		2443			
	SUB-TOT		36			24917	4426		
	STF		41	67.5	FTE	11475			
	##		41	0.0		532			
	SUB-TOT		41			12007	6375		
	STF		44	24.8	FTE	4216	2096		
	STF		46	109.0	FTE	18530			
	##		46	0.0		2845			
	SUB-TOT		46			21375	12945		
	STF		51	187.9	FTE	31943	27328		
	STF		53	30.2	FTE	5134	9921		
	STF		54	10.0	FTE	1700	2891		
	STF		55	402.6	FTE	68442	8295		
	STF		61	16.2	FTE	2754	5673		
	STF		62	43.0	FTE	7310	10408		
	STF		63	66.0	FTE	11220			
	##		63	0.0		10278			
	SUB-TOT		63			21498	18421		
	STF		64	42.5	FTE	7225	1484		
	STF		65	320.6	FTE	54502	8107		
	STF		67	10.2	FTE	1353	5408		
	##		71	0.0		146	146		
	##		72	0.0		501	501		
	TOT					609576	413752	-195824	-47
STUDY FACILITIES	410	##	11	0.0		13849	13849		
		##	22	0.0		2535	2535		
		##	35	0.0		163	163		
		TSD	41	3665.0	FTE	109950			
		##	41	0.0		2194			
		SUB-TOT	41			112144	64372		
		##	46	0.0		1969	1969		
		##	53	0.0		394	394		
		TOT				131054	83282	-47772	-57
STACK SPACE	420	##	11	0.0		1710	1710		
		##	22	0.0		133	133		
			41	806.3	VOL	70148	74706		
		##	46	0.0		3019	3019		
		##	55	0.0		163	163		
		TOT				75173	79731	4558	5
LIBRARY PROCESSING	440	##	41	0.0		14463	14463	0	0
SPECIAL USE FAC	500	##	11	0.0		111401	111401		
		##	35	0.0		2927	2927		
		##	36	0.0		22997	22997		
		##	51	0.0		25454	25454		
		##	64	0.0		1870	1870		
		##	71	0.0		65	65		

Table 3.4. (continued)

KANSAS STATE UNIVERSITY : SPACE SUMMARY BY ROOM TYPES.

DESCRIPTION	ROOM-TYPE		POS	ACTIVITY UNIT LOAD	PROJECTED NASF	EXISTING NASF	SURPLUS/ DEFICIT	% OF EXIS
	TOT				164914	164914	0	0
GENERAL USE FAC	600	##	11	0.0	8723	8723		
		##	22	0.0	8626	8626		
		##	32	0.0	5406	5406		
		##	35	0.0	4244	4244		
		##	36	0.0	3148	3148		
		##	41	0.0	210	210		
		##	46	0.0	223	223		
		##	51	0.0	161802	161802		
		##	55	0.0	45844	45844		
		##	62	0.0	111	111		
		##	67	0.0	111	111		
		##	71	0.0	1340	1340		
	TOT				239788	239788	0	0
SUPPORTING FAC	700	##	11	0.0	17327	17327		
		##	22	0.0	8209	8209		
		##	35	0.0	4379	4379		
		##	41	0.0	1689	1689		
		##	44	0.0	760	760		
		##	46	0.0	535	535		
		##	51	0.0	6448	6448		
		##	53	0.0	383	383		
		##	55	0.0	6162	6162		
		##	63	0.0	1545	1545		
		##	64	0.0	10260	10260		
		##	65	0.0	38398	38398		
		##	67	0.0	36	36		
		##	71	0.0	11534	11534		
		##	72	0.0	523	523		
	TOT				108188	108188	0	0
MEDICAL CARE FAC	800	##	11	0.0	13359	13359		
		##	22	0.0	537	537		
	TOT				13896	13896	0	0
RESIDENTIAL	900	##	11	0.0	397	397		
		##	22	0.0	65	65		
		##	36	0.0	1587	1587		
		##	41	0.0	3366	3366		
		##	51	0.0	47448	47448		
		##	63	0.0	5223	5223		
		##	65	0.0	744	744		
		##	71	0.0	790020	790020		
		##	81	0.0	5834	5834		
	TOT				854684	854684	0	0
UNASSIGNABLE, ETC	888	##	81	0.0	8712	8712	0	0
GRAND-TOTAL					3576251	2866424	-709827	-24

KANSAS STATE UNIVERSITY : SPACE SUMMARY BY PCS

PCS	DESCRIPTION	PROJECTED NASF	EXISTING NASF	SURPLUS/ DEFECIT	% OF EXIS
10	INSTRUCTION				
11	GEN ACADEMIC INSTR	1107023	830125	-276898	-33
	SUB-TOTAL	1107023	830125	-276898	-33
20	ORGANIZED RESEARCH				
21	INSTITUTES & RES CTR	18223	3594	-14629	-407
22	INDIV OR PROJ RES	697513	482956	-214557	-44
	SUB-TOTAL	715736	486550	-229186	-47
30	PUBLIC SERVICE				
31	COMMUNITY EDUCATION	9350	3756	-5594	-148
32	COMMUNITY SERVICE	6544	6544	0	0
33	CO-OP EXTEN SERVICE	51		-51	*
35		54952	41266	-13686	-33
36	UNIDENTIFIED	62568	42077	-20491	-48
	SUB-TOTAL	133465	93643	-39822	-42
40	ACADEMIC SUPPORT				
41	LIBRARIES	214027	165181	-48846	-29
44	COMPUTING SUPPORT	4976	2856	-2120	-74
46	ACAD ADMN & PER DEV	27121	18691	-8430	-45
	SUB-TOTAL	246124	186728	-59396	-31
50	STUDENT SERVICE				
51	SOCIAL & CUL DEV	273295	268680	-4615	-1
53	COUNSEL & CARI GUID	5911	10598	4687	44
54	FINANCIAL AID	1700	2891	1191	41
55	STUDENT SUPPORT	120611	60464	-60147	-99
	SUB-TOTAL	401517	342633	-58884	-17
60	INST - SUPPORT				
61	EXECUTIVE MGMT	2754	5673	2919	51
62	FISCAL OPERATIONS	7421	10519	3098	29
63	GEN ADMN SERVICES	28529	25452	-3077	-12
64	LOGISTICAL SERVICES	19355	13614	-5741	-42
65	PHYSICAL PLANT OPER	93644	47249	-46395	-98
67	COMMUNITY RELATIONS	2000	5555	3555	63
	SUB-TOTAL	153703	108062	-45641	-42
70	INDEP - OPERATIONS				
71	INSTITUTIONAL OPER	803113	803113	0	0
72	OUTSIDE AGENCIES	1024	1024	0	0
	SUB-TOTAL	804137	804137	0	0
80	MISCELLANEOUS				
81	UNASSIGNABLE, ETC	14546	14546	0	0
	SUB-TOTAL	14546	14546	0	0
	TOTAL	3576251	2866424	-709827	-24

Table 3.5

been shown. It is interesting to note that the overall totals shown in this report and in the university space summary report tally, both representing the total space existing (and projected) for the entire university.

3.3 DATA ELEMENTS.

3.3.1 GENERAL.

Once the format of the planning reports have been finalized, the next major step in the planning process is the proper organization of the different data files to facilitate easier generation of these reports. Since the information in these files will be subjected to periodical changes, there should be an efficient and well defined inventory gathering procedure. Also the data files should lend themselves to easy updating, reporting, and rearranging.

For the planning work the following data are essential:

a) Activity Load: WCH, Personnel, etc.,

b) Existing Space.

c) Space Factors.

The activity load and the existing space data are essentially independent of each other. They are obtained from entirely different sources, and hence it is only appropriate that they be stored in different files. The space factors can be stored in the activity load data file. However it was found preferable to read in the space factors from the input card file.

For convenience of analysis each room type described in Section 2.4 was allotted a two digit code. These codes run from 01 to 13, one for each of the thirteen major room types, shown in Table 3.6. The sections that follow will present a general introduction to these files. For more details such

Table 3.6

Room Type - Room Code Relation

ROOM TYPE DEFINITION	ROOM-TYPE	ROOM CODE
CLASSROOM	100	1
CLASS LABORATORY	200	2
NON CLASS LABORATORY	250	3
OFFICE	300	4
STUDY	410	5
STACK	420	6
LIBRARY PROCESSING	440	7
SPECIAL USE FACILITIES	500	8
GENERAL USE FACILITIES	600	9
SUPPORTING FACILITIES	700	10
MEDICAL CARE FACILITIES	800	11
RESIDENTIAL	900	12
MISCELLANEOUS, UNASSIGNABLE, ETC.	888	13

as the computer programs, card formats etc., the reader is referred to Chapter 5.

3.3.2 ACTIVITY LOAD DATA FILE.

The activity load data essentially consist of the WCH, personnel data, and data such as total volumes, etc., which are used for projecting some supporting facilities. The WCH data are derived from the room utilization studies done at Kansas State University. The personnel data come principally from two sources: the university budget, and the files maintained by the Office of Admissions and Records. These data were put together by the office of Vice President for Planning, and Figure 3.1 shows a sample sheet from their file. Table 3.7 shows the activity load and space factor coding sheet developed by the author. For each department on campus, this sheet has the following information: Hegis, room types (and room codes), associated PCS, load factor, activity load, and space factor. The load factor (two digit code) is essentially an indicator which points to a particular data type in the coding sheet. The file has been set up as a permanent disk data set. Each department on campus has a record in this file. The department code serves as the key for each record. Each record contains the activity load data needed to project space types most commonly encountered. These include:

1) Classroom - Type 100;
2) Class laboratory - Type 200;
3) Non class laboratory - Type 250;
4) Office - Type 300. Two of the following: Instruction, Research, Public Service, Extension, Support Function;
5) Study facilities - Type 410.

DEPARTMENT AGRICULTURAL ECONOMICS CODE 2.2%

HEGIS CODE 1.1.0111
2.2.0111

PCS	Program Category	Load Unit		Load Total	Space Factor	Space Required	Space Available	Deficit	Surplus	Existing Space Conditions			
										Satisfactory	R1	R2	Obsolete
1.1	Instruction	WSH		1782	1.0	1782							
1.1	Instruction	WSH	L	0	3.4	0							
			U	210	3.4	714.0							
			G	16									
			T	226									
2.2	Research	HC	F	19	1300	23400							
			S	11	250	2750							
			RA	11.7	1320	15512							
			T			41360	580				580		
1.1	Instruction	FTEF		8.0	170	1360.0							
		FTES		1.5	170	255.0							
2.2	Research	FTEF		22.9	170	3893.0							
		FTES		10.5	170	1785.0							
3.0	Public Service	FTEF		0.4	170	38[illegible].0							
		FTES		10.0	170	1700.0							
4.0	Academic Support	FTE			170								
5.0	Student Services	FTE			170								
6.0	Institutional Support	FTE			170								
	Total	FTE				12811	12408				12808		
							261				261		
							522				522		
1.1	Total Instruction												
2.2	Total Research												

46

Figure 3.1 Sample sheet from the data file maintained by the office of Vice President for Planning.

Table 3.7

Activity Load and Space Factor Coding Form

DEPT-NAME: DEPT-CODE:

COLLEGE: HEGIS:

SPACE TYPE	DATA-DESCRIP	DATA	LOAD FACTOR	SPACE FACTOR	PCS
CLASSROOM (100)	WCH		1		
CLASSROOM LABORATORY (200)	WCH FS JS G1 G2		2 3 4 5		
STUDY (410)	FTE FS JS G1 G2 TOT		6 7 8 9 10		
NON CLASS LAB (250	HC G1 G2 FAC STF		11 12 13 14		
OFFICE (INSTRUCTION)	FTE FAC GR STF		15 16 17		
OFFICE (INSTRUCTION,	FTE FAC GR STF		18 19 20		
STUDY, ETC.	FTE FAC		21		
STACK (420)	VOL (thousands)		22		
RESIDENTIAL (900)	HC SIN MAR		23 24		
SPECIAL USE, GENERAL USE, SUPPORT FAC,ETC.	IF SPEC AS A PERCENT OF OTHER SPACE TYPES				
OTHER (SPECIFY)					

Each activity load data type has an associated indicator called the Load Factor (Table 3.7). Each record can be best described by the following structure (Appendix C):

```
1    DEMAND-DATA,
     2    HEGIS,
     2    DEPT-CODE,
     2    DEPT-NAME,
     2    DATA-TYPE (21);
```

The load factor will indicate the computer program, the proper activity load (data type) to be used. The activity load information which is not stored in this record is read in from the input file. The reader is referred to Section 5.2.3 for more detail.

3.3.3 EXISTING SPACE INVENTORY FILE.

An existing space inventory file is being maintained by the university. For each department on campus the existing space is available by room type (all minor breakdowns) and function (Section 5.2.1). For the facility planning work, it is desirable to have these data by room types (only major breakdowns) and PCS. Hence this information had to be processed (Section 5.2.1) prior to setting up this file. The file has been set up as a disk data set. Each department on campus has a record in this file. The department code serves as the key for each record. The record stores all room codes (hence room types), the associated PCS and the existing net assignable square feet of space. Each record can be best described by the following structure (Appendix C):

```
1     EXISTING-SPACE,
   2     HEGIS,
   2     DEPT-CODE,
   2     DATA(14),
      3     ROOM-CODE,
      3     PCS,
      3     NASF;
```

From above it is clear that each record stores the existing space data for fourteen different combinations of room types and PCS, which is about the maximum that was encountered. Refer to Section 5.2 for further details.

3.3.4 INPUT FILE.

Once the activity load and the existing space inventory files have been set up, the following information has to be read into the computer, from a separate file, so as to trigger the projection mechanism into action:

a) Identifying information of the department (department code) for which space projection is sought.

b) Room type code, associated PCS, load factor code, and space factor. Each set of this information will generate the printing of one line in the department space summary report.

c) Any activity load data not stored in the Activity Load Data File.

The input file was set up as a card file, sequenced on the department code. A constant reference to both the activity load data and the existing space inventory data is to be made, while setting up this file. Care should be taken to see that no room types, PCS, or load factor information are missed. The reader is referred to the User's Manual (Appendix B) for additional information.

3.4 LOGIC DIAGRAM FOR OBTAINING PLANNING REPORTS.

The computer logic diagram shown in Figure 3.2 exemplifies the route to be followed in obtaining the different summary reports.

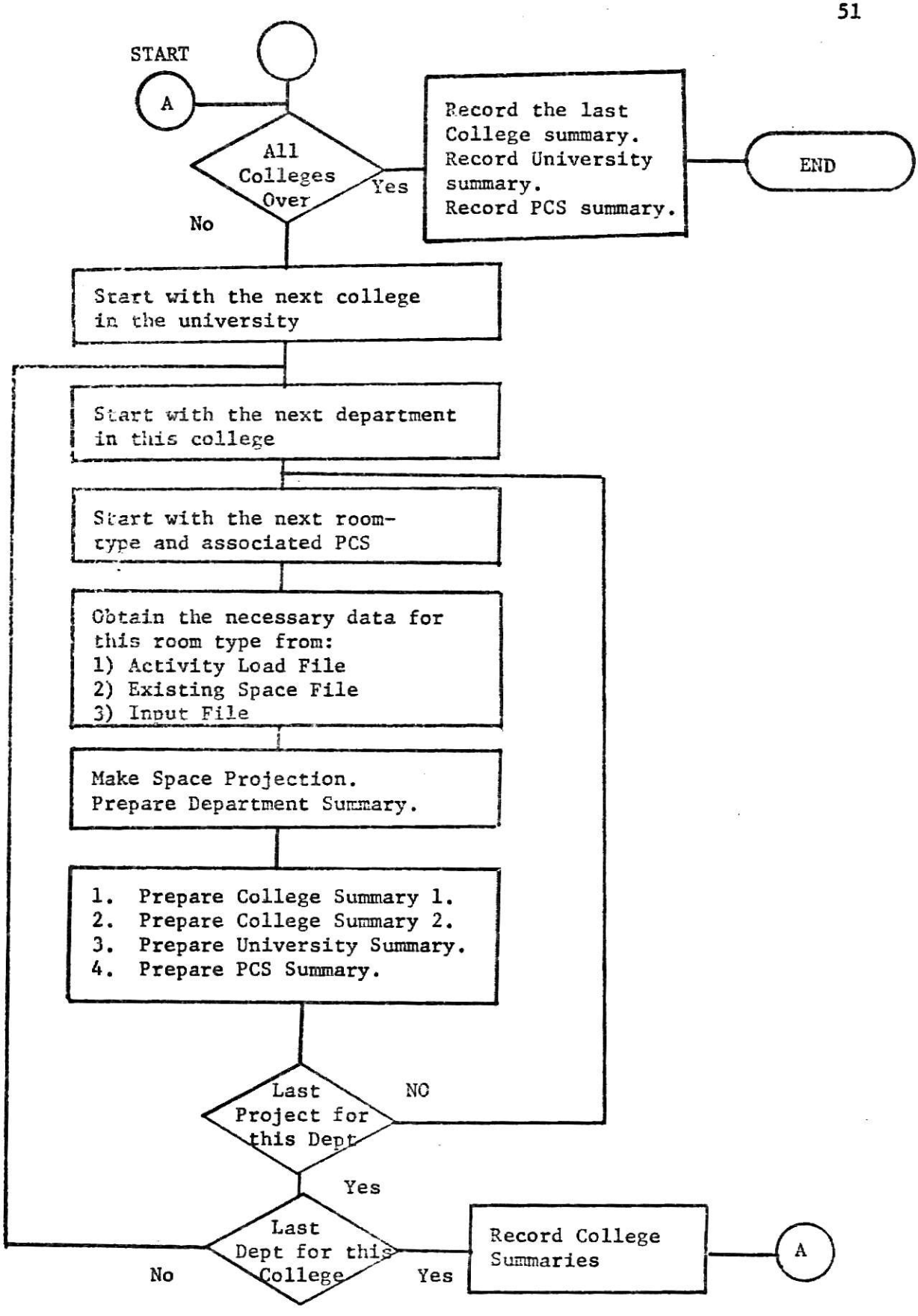


Figure 3.2. Logic Diagram for Obtaining Planning Reports.

CHAPTER 4

SUMMARY

4.1 INTRODUCTION.

The main objective of this work was to develop data processing programs for facilities planning. The effort was planned to eliminate, if possible, the few shortcomings of CRS work. The main theme of this chapter will be a comparative study of this work with that of CRS. This chapter will bring out the shortcomings of the present work, and suggest some possible future improvements.

4.2 NEW SYSTEM AND OLD SYSTEM - A COMPARATIVE STUDY.

The present work differs from that of CRS in many aspects. However this effort has its own shortcomings, brought to light in Section 4.3. Hence no attempt will be made to make any quantitative comparison of the results obtained, with those obtained by CRS. Only a qualitative comparison will be attempted. The following paragraphs will highlight the areas in which the present work differs from that of CRS.

FORMAT OF SUMMARY REPORTS: Planning reports have been summarized on the different organizational units, instead of on the program classification structure used by CRS. The reports are more elaborate and detailed than those presented by CRS.

PROJECTION DETAILS: A few minor changes were made in the projection details of some room types. For room types 100 and 200 CRS used the weekly student hour data for the activity load. The weekly student hour data were derived from the credit hour data, using an approximate transformation ratio. However, the best approach would be to use the WCH, which involves no

approximation whatsoever. The WCH data were available for Kansas State University, and use was made of them in the present work. For simplicity of calculation, CRS used a weighted mean of space factors, while projecting space for room types that have subcategories (such as type 200). This has been avoided in the present work. In some cases, CRS has attempted to project space for the supporting, general use, special use, and medical care facilities. The activity load and space factors used are questionable. There is no documentation on the approach used. No attempt has been made to project space for these room types, in the present work.

4.3 SHORTCOMINGS OF THIS WORK.

4.3.1 PROJECTION PROCEDURE.

Very little study was done on room types 500, 600, 700, 800 and 900; and hence no projection procedure has been developed. For the reasons mentioned in Section 2.4, the projected space has been set equal to the existing space in such cases. A similar thing has been done with any other room type for which there was no activity load data for making projection. The above cases have been flagged with a '##' sign in the summary reports. Spaces with no assigned activity load may represent classification errors in the existing space, thus no projection should be made. The current procedure inflates the projected space if this condition exists. More attention should be focussed on these areas, in the future revisions of this work.

4.3.2 SPACE CLASSIFICATION BY PCS.

In the present work there might be many instances found where spaces would have been misclassified, as far as the PCS sub-category is concerned,

for the following reason. The existing space inventory maintained by the university is by room type and associated function. Table 5.1 is the list of function codes. It was found that there is no one-to-one correspondence between the function codes and PCS. Hence the translation of these function codes into PCS is approximate. In the final stage, while preparing data for the input file, the author noticed many instances wherein the NASF for a particular room type and PCS existed, but there was no activity load data for making a projection, and vice versa. To avoid this difficulty, either the existing space inventory data maintained by the university has to be revised, by making a fresh survey where PCS codes are used instead of function codes; or a clearcut relationship between function codes and PCS codes has to be defined.

4.3.3 DEPARTMENT CODES.

In this study an effort was made to develop a new set of identification codes for the organizational units. It is intended that these codes, in addition to identifying college affiliation, also identify the main function affiliation of the different organizational units. However there are a few departments, such as Office of the Dean of Engineering (Dept code=4401), even though being academic support in nature, have predominantly instruction and/or research activity. In this case the reason is that this department offers courses in General Engineering. Hence this department was affiliated to the College of Engineering even though in theory it should be affiliated to the college with Academic Support activity. Due to these relatively few irregularities, the department codes within a particular college, are not always truly sequential.

4.3.4 DATA PROCESSING.

The data processing programs have been presented in Chapter 5. If any further revision of the program is made, the author recommends that the input requirements be (format of INPUT1 file) made less rigid. For a particular department, the room types, associated PCS, and the load factor code information must be presented in the same sequence in which the department summary is required. 'List Processing' features of PL/I could be incorporated to obviate this difficulty.

CHAPTER 5

DATA PROCESSING PROGRAMS

5.1 INTRODUCTION.

One of the important objectives of this report is the development of a Data Processing procedure for facilities planning work. With the variety and volume of data handled, it is imperative that electronic data processing equipment be used, for speed of production and dissemination of information. The important decision to be made in developing this data processing procedure is the selection of the computer programming language most appropriate for this kind of application. The language chosen should have extensive file handling and character handling capabilities. The obvious choice is either PL/I or Cobol. Because PL/I has more extensive character handling and input/output features, it was chosen.

5.2 EXISTING SPACE INVENTORY FILE SET UP.

Since most of the output reports summarize space on room type major and PCS minor breakdowns, it is best to store the existing space inventory for each department by room type major and PCS minor breakdowns. Each department on campus will have a record in this file, and the department code will be the key for each record. A record key identifies the information used by the operating system to locate the proper record from the data set. For each department there is a provision to store existing space data for fourteen different combinations of room types and PCS, which is about the maximum that was encountered. If this has to be enlarged, the file should be set up again from the beginning. The dimension of the different storage arrays and the logical record length must be changed.

5.2.1 PROCESS.

OBJECTIVE: The program PROCESS was written to process the existing space inventory data maintained by the university. For each department the existing space is by room type (all minor breakdowns) and function. Table 5.1 is the list of function codes. The output of this program would be existing space summarized by room type (major breakdowns only) and function.

DESCRIPTION: Firgure A.1 is the flow chart for this program. Figure A.2 is the source listing and job control requirements. Figure A.3 is the input card format. The output of this program is the NASF, summarized by room types (major breakdowns only) and function. The output was punched on cards. After changing the function codes into PCS codes, these cards were used as input to the program SETUP1, which sets up the existing space inventory file on disk. Figure A.4 is the card format of the punched output.

NOTE: The author is unable to define any rigid set of rules for converting the function codes to PCS codes. The decisions made were subjective and in many cases arbitrary. Section 4.3 of this report has more detailed discussion on this aspect.

5.2.2 SETUP1.

OBJECTIVE: Program SETUP1 was written to set up the existing space inventory file as a permanent data set on disk.

DESCRIPTION: The file is of the Regional (3) type (Appendix C). The department code is the comparison key. Figure A.5 is the flow chart. Figure A.6 is the source listings and job control required. Figure A.4 is the input card format.

Table 5.1

List of Function Codes

10 INSTRUCTION

20 RESEARCH

30 PUBLIC SERVICE

40 LIBRARY

50 GEN ADMN AND INSTITUTIONAL SERVICES

60 AUXILARY SERVICE

70 NON INSTITUTIONAL AGENCIES

80 UNASSIGNED AREA

- 81 Inactive Area
- 82 Alteration or Conversion Area
- 83 Unfinished Area

99 PROBATE

00 NON ASSIGNABLE AREA

- 01 Custodial Area
- 02 Circulation Area
- 03 Mechanical Area
- 04 Construction Area

5.2.3 UPDATES.

OBJECTIVE: Program UPDATES has been written to provide for future updating of the existing space inventory file.

DESCRIPTION: This program can be used to add new records, update existing records, or delete obsolete records. The input card format is shown in Figure A.4. A blank in the first field indicates that it is a new record to be added to the file, 'U' indicates that the record is to be updated, and 'D' signifies that the record is to be deleted. Figure A.7 is the flow chart, and Figure A.8 is the source listing and job control required.

5.3 ACTIVITY LOAD DATA FILE SET UP.

Each department on campus has a record in this file and the department code serves as a key to each record. Each record in this file can be best described by the following PL/1 structure (Appendix C):

```
DCL  1  DEMAND_DATA,  2  HEGIS  CHAR(4),
        2  ADM_CODE  CHAR(4), 2 DATA_TYPE(21) DEC FIXED(5,1);
```

The subscripts of the DATA_TYPE indicate the corresponding load factors. As seen the records in this file can only store activity load data corresponding to load factors 1 thru 21. Data with load factors other than these, and also data with load factors 1 thru 21 occuring more than once, must be read from the input file. Only rarely does this situation arise. To make efficient use of the direct access storage, this method has been implemented. More information regarding the proper transmission of the activity load data from the input file is given in the User's Manual (Appendix B).

5.3.1 SETUP2

OBJECTIVE: This program was written to set up the activity load data file.

DESCRIPTION: The file is of the Regional (3) type (Appendix C). The department code is the comparison key. Figure A.9 is the flow chart, and Figure A.10 is the source listing and job control required. Figure A.11 shows the input card format.

5.3.2 UPDATED.

OBJECTIVE: This program was written to provide for future updating of the activity load data file.

DESCRIPTION: This program can be used to add new records, update existing records, or delete obsolete records. The input card format is shown in Figure A.11. A blank in the first field indicates that it is a new record to be added to the file, 'U' indicates that the record is to be updated, 'D' indicates that the record is to be deleted. Figure A.12 is the flow chart, and Figure A.13 is the source listing and job control required.

5.4 SPACE FACTOR COMPUTATION.

Two short programs were written to compute space factors for class laboratories and non-class laboratories. The projection procedure for the class laboratory category was different from that of CRS. Hence the space factors used by CRS could not be borrowed. However for the non-class laboratory the space factors computed here were not used (Section 2.4). The space factors for this space type were the guideline values given in WICHE, used by CRS.

5.4.1 SFACTOR1.

This program computes the space factor for the class laboratory by the three student levels--lower undergraduates (freshmen and sophomores), upper undergraduates (junior and senior), and graduates. The program reads in the square feet per station, room utilization rate, and station occupancy ratio, then computes the space factor using the relation

Space Factor = Square feet per station/(RUR*SOR)

Figure A.14 is the source listing, and Table 2.1 is the output of this program.

5.4.2 SFACTOR2.

This program computes the space factor for the non-calss laboratory type, using the method suggested by Bareither (1) described in Section 2.4 of this report. Figure A.15 is the source listing and job control required. Table 2.3 is the output of this program.

5.5 MAIN PROGRAM FOR FACILITIES PLANNING.

The entire program developed for facilities planning consists of one main program and five external subroutines. Description of each module follows.

5.5.1 SPPROJ.

This is the main controlling procedure. It accomplishes the following functions:

a) Reads in all input,

b) Processes the input,

c) Prepares and prints the department space summary,

d) Updates PCS totals,

e) Calls procedures TITLES, ERRORM, COLPRNT, UNVPRNT, and PCSPRNT, each of which performing a specific function.

Figure A.16 is the flow chart, and Figure A.17 is the source listing. The program has been thoroughly documented.

5.5.2 COLPRNT.

This is an external procedure (Appendix C). It updates all college totals and prints them whenever the college changes. Figure A.19 is the source listing. The program has been documented thoroughly.

5.5.3 UNVPRNT.

This is an external procedure. It updates all university totals and prints them after all the input information has been processed. Figure A.20 is the flow chart, and Figure A.21 is the source listing. The program has been documented thoroughly.

5.5.4 PCSPRNT.

This is an external procedure. It prints the PCS totals after all the input information has been processed. Figure A.22 is the flow chart and Figure A.23 is the source listing. The program has been thoroughly documented.

5.5.5 ERRORM.

This is an external procedure called by other procedures whenever an error is detected. This procedure gives appropriate diagnostics. The error message provides the programmer with information such as the procedure in which that error was encountered, and the probable cause for that error

ERROR CONDITION	PROCEDURE	ACTION
Input cards out of sequence	MAIN	Execution terminated
Improper Load Factor Code	MAIN	Input data corresponding to this Load Factor is ignored.
Improper PCS code	MAIN	Input data corresponding to this PCS code is ignored.
Record missing in either the Activity Load File or the Existing Space Inventory File	MAIN	Input information for that department is ignored.
Data Conversion	MAIN	Input data is ignored.
Subscriptrange	MAIN, COLDPRNT, UNVPRNT	If ISTOP=0 then the execution is terminated, otherwise the execution is allowed to continue (debugging runs).

Figure 5.1 Error Conditions taken care of in the Computer Program

condition. If the error is of a severe type (subscriptrange, etc.) the execution is terminated. Figure 5.1 lists the error conditions that have been taken care of, and the procedures in which that error condition is detected. Figure A.24 is the source listing.

5.5.6 TITLES.

This is an external procedure which prints the necessary information on the title page. This procedure can be revised to provide any additional information deemed necessary. Figure A.25 is the source listings.

REFERENCES

1. Bareither, H. D., and Schillinger, J. L., University Space Planning, University of Illinois, Urbana, Chicago, London, 1968.

2. Sumner, N. L., Socio Economic Planning Sciences, Vol. 2, April 1969.

3. Western Interstate Commission for Higher Education; Higher Education Facilities Planning and Management Manuals, Vol. 1-7, May 1971.

4. Richard, W. J., and Jack, B. L., A New Tool for Educational Administrators, Published for the Association of Universities and Colleges of Canada by University of Toronto Press, Canada, 1965.

5. Van Wijk, A. P., Russel, R. S., and Atcheson, R. M., The use of Simulation Models in Educational Planning, Paper presented at: Canadian Operations Research Society, 1971 Annual Conference "O. R. in the public sector", Ottawa, Canada, June 9, 10, 11, 1971.

6. Claudill, Rowlett and Scott, (consultants): Physical Development Planning Manual, March 1972.

7. Leonard, C. R., Higher Facilities Inventory and Classification Manual, (Draft for prepublication review).

APPENDIX A

Program Listings, Job Control required,

Flow Charts, and Input Card Formats.

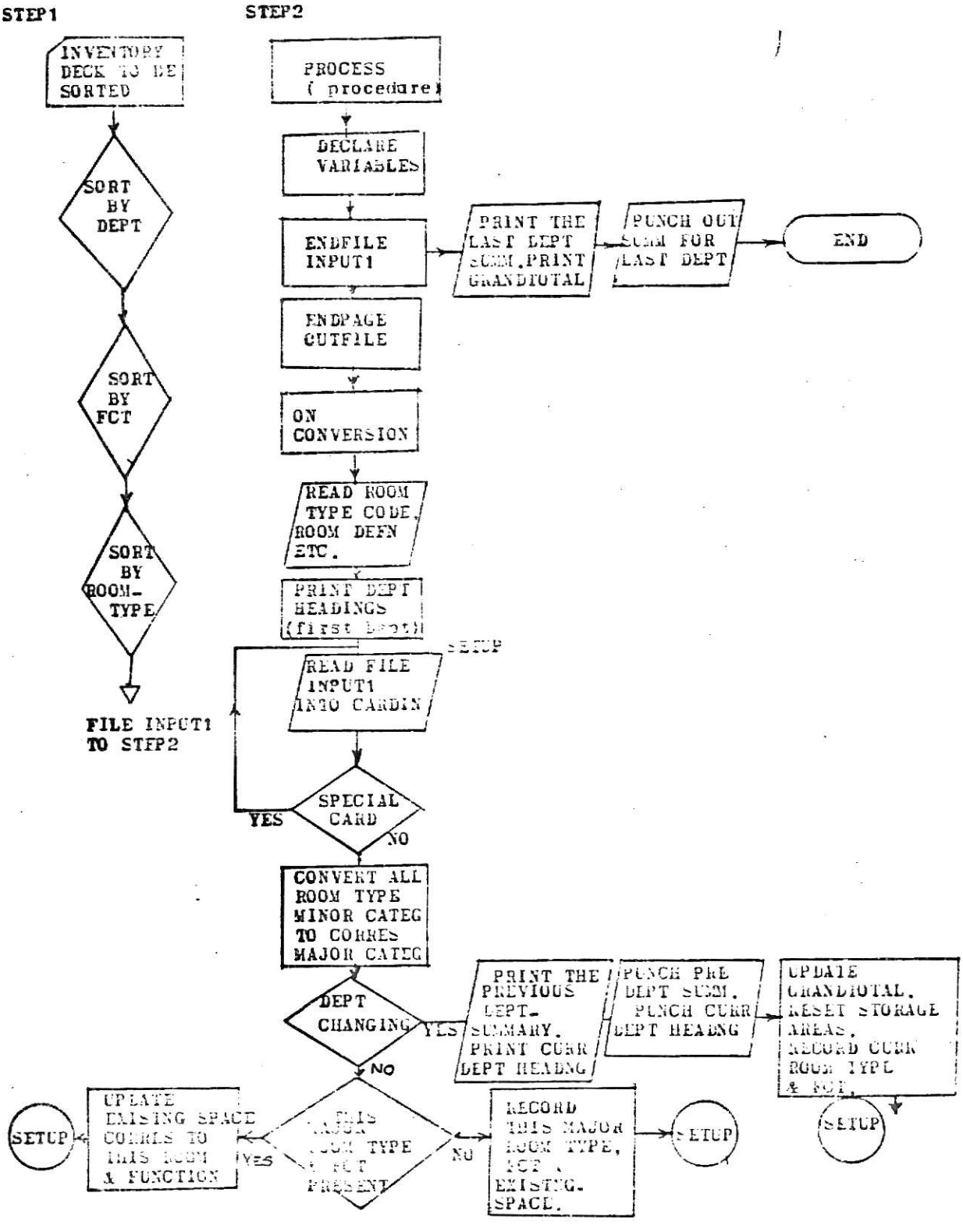


Figure A.1. Flow Chart of Program PROCESS

Figure A.2. PROCESS: Source Listings and Job Control required

```
KSU'S PL/I BEATENER AND PRECOMPILER                    PAGE

PROCESS:PROCEDURE OPTIONS(MAIN);                                          1    1
/*                                                                      */  1    2
/*  ********************************************************************/  1    3
/*  THIS PROGRAM PROCESSES THE EXISTING SPACE INVENTORY MAINTAINED-     */  1    4
/*  BY THE UNIVERSITY                                                   */  1    5
/*  ..................... DESCRIPTION OF IMPORTANT VARIABLES ......     */  1    6
/*  SPDATA : THIS IS A STRUCTURE WHICH STORES 15 COMBINATIONS OF        */  1    7
/*  TYPES, FUNCTION AND CORRESPONDING EXISTING SPACE (A DEPARTMENT)     */  1    8
/*  CARDIN : BASED, CHAR(80) VARIABLE INTO WHICH INPUT IS READ          */  1    9
/*  GRANDTOTAL : TOTALS EXISTING SPACE FOR THE ENTIRE UNIVERSITY        */  1   10
/*  TOTAL :  TOTALS EXISTING SPACE FOR A DEPARTMENT                     */  1   11
/*                                                                      */  1   12
/*  ********************************************************************/  1   13
                                                                          DCL   1   14
BEGIS CHAR(4), DEPT_NAME CHAR(20), ROOM_DEF(13) CHAR(3),                  DES C 1   14
RIPTION(13) CHAR(20);                                                     1   14
        DCL (STDEPT_CODE,DEPT_CODE) CHAR(4),                              1   15
        FUNCTIONIN  FIXED BIN,      ROOM_TYPE FIXED BIN,                  1   15
        EXSPACEIN  FIXED BIN(31)  INIT(0);                                1   15
        DCL TOTAL BIN FIXED(31) INIT(0),   GRANDTOTAL  BIN FIXED(47) IN   1   16
IT(0);                                                                    1   16
        DCL CARDIN CHAR(80) BASED (P);                                    1   17
        DCL ROOM_CODEIN  FIXED BIN;                                       1   18
        DCL DEPT_CODE1  CHAR(4);                                          1   19
        DCL 1 SPDATA(15), 2 ROOM_CODE FIXED BIN, 2 FUNCTION FIXED BIN,    1   20
        2  EXSPACE  FIXED BIN(31);                                        1   20
        ON ENDFILE(SYSIN)                                                 1   21
        DEPT_CODE1=DEPT_CODE;                                             1   22
        ON ENDFILE(INPUT1)                                                1   23
           BEGIN;                                                         2   24
           CALL PRINT;                                                    2   25
           GRANDTOTAL=GRANDTOTAL+TOTAL;                                   2   26
           PUT SKIP(4)FILE(OUTFILE)EDIT('GRAND-TOTAL',GRANDTOTAL)(COL(75  2   27
           ),A,COL(94),P'(12)Z');                                         2   27
           PUT SKIP(3)FILE(OUTFILE)EDIT('NORMAL TERMINATION OF THE PROGR  2   28
           AM')(COL(5),A);                                                2   28
           GO TO END;                                                     2   29
           END;                                                           2   30
        ON ENDPAGE(OUTFILE)                                               1   31
           BEGIN;                                                         2   32
           PUT PAGE FILE(OUTFILE)EDIT('KANSAS STATE UNIVERSITY  : EXISTI  2   33
           NG SPACE SUMMARY BY ROOM TYPES AND FUNCTION')(COL(35),A);      2   33
           PUT SKIP(2)FILE(OUTFILE)EDIT('DEPT- BEGIS   DEPARTMENT-NAME    2   34
                 DESCRIPTION      ROOM  FCT   EXISTING')(COL(31),A)('CO   2   34
           DE','TYPE','NASF')(COL(31),A,COL(86),A,COL(102),A);            2   34
           END;                                                           2   35
        ON CONVERSION                                                     1   36
           BEGIN;                                                         2   37
           PUT SKIP EDIT('ERROR IN DATA CARD:  ',CARDIN)(A);              2   38
           GO TO SETUP;                                                   2   39
           END;                                                           2   40
        OPEN FILE(OUTFILE)PRINT PAGESIZE(59)LINESIZE(132);                1   41
        OPEN FILE(PUNCH)STREAM OUTPUT;                                    1   42
        SIGNAL ENDPAGE(OUTFILE);                                          1   43
        GET FILE(SYSIN)EDIT((ROOM_DEF(I) DO I=1 TO 13))(X(1),A(3));       1   44
        GET SKIP FILE(SYSIN)EDIT(DESCRIPTION)(A(20));                     1   45
        GET SKIP FILE(SYSIN);                                             1   46
        OPEN FILE(INPUT1)RECORD SEQUENTIAL INPUT;                         1   47
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

        STDEPT_CODE='    ';                                                    1   36
        SPDATA=0;                                                              1   37
        GET FILE(SYSIN)EDIT(HEGIS,DEPT_NAME)(X(15),A(4),X(1),A(20));          1   38
        GET SKIP FILE(SYSIN);                                                  1   39
        PUT SKIP(2)FILE(OUTFILE)EDIT(STDEPT_CODE,HEGIS,DEPT_NAME)(COL(31       1   40
        ),A,COL(37),A,COL(43),A);                                              1   40
SETUP:  READ FILE(IN,PUT1)SET(P);                                              1   41
           IF SUBSTR(CARDIN,80,1)¬=' ' THEN                                    2   42
           GO TO SETUP;                                                        2   43
           IF SUBSTR(CARDIN,54,1)='T' THEN                                     2   44
           GO TO SETUP;                                                        2   45
        GET STRING(SUBSTR(CARDIN,32,2))EDIT(FUNCTIONIN)(F(2));                 1   46
           IF FUNCTIONIN<10 THEN                                               2   47
           GO TO SETUP;                                                        2   48
           IF FUNCTIONIN=99 THEN                                               2   49
           GO TO SETUP;                                                        2   50
        DEPT_CODE=SUBSTR(CARDIN,60,4);                                         1   51
           IF DEPT_CODE='CLSR' THEN                                            2   52
           DEPT_CODE=' 909';                                                   2   53
           IF DEPT_CODE='CRUN' THEN                                            2   54
           DEPT_CODE=' 909';                                                   2   55
           IF DEPT_CODE='NKSU' THEN                                            2   56
           DEPT_CODE=' 910';                                                   2   57
           IF DEPT_CODE='RESD' THEN                                            2   58
           DEPT_CODE=' 911';                                                   2   59
        GET STRING(SUBSTR(CARDIN,49,3))EDIT(ROOM_TYPE)(F(3));                  1   60
           IF ROOM_TYPE>499 THEN                                               2   61
           GO TO TYPE2;                                                        2   62
           IF ROOM_TYPE>99 THEN                                                2   63
              IF ROOM_TYPE<250 THEN                                            3   64
                 DO;                                                           4   65
                 ROOM_CODEIN=ROOM_TYPE/100;                                    4   66
                 GO TO SEARCH;                                                 4   67
                 END;                                                          4   68
           IF ROOM_TYPE>249 THEN                                               2   69
              IF ROOM_TYPE<399 THEN                                            3   70
                 DO;                                                           4   71
                 ROOM_CODEIN=ROOM_TYPE/100+1;                                  4   72
                 GO TO SEARCH;                                                 4   73
                 END;                                                          4   74
           IF ROOM_TYPE=410 THEN                                               2   75
              DO;                                                              3   76
              ROOM_CODEIN=5;                                                   3   77
              GO TO SEARCH;                                                    3   78
              END;                                                             3   79
           ELSE                                                                2   80
              IF ROOM_TYPE=455 THEN                                            3   80
                 DO;                                                           4   81
                 ROOM_CODEIN=5;                                                4   82
                 GO TO SEARCH;                                                 4   83
                 END;                                                          4   84
           IF ROOM_TYPE=420 THEN                                               2   85
              DO;                                                              3   86
              ROOM_CODEIN=6;                                                   3   87
              GO TO SEARCH;                                                    3   88
              END;                                                             3   89
           ELSE                                                                2   90
              IF ROOM_TYPE=430 THEN                                            3   90
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

                  DO;                                                    4   91
                  ROOM_CODEIN=6;                                         4   92
                  GO TO SEARCH;                                          4   93
                  END;                                                   4   94
            IF ROOM_TYPE=440 THEN                                        2   95
               DO;                                                       3   96
               ROOM_CODEIN=7;                                            3   97
               GO TO SEARCH;                                             3   98
               END;                                                      3   99
TYPE2:   ROOM_CODEIN=ROOM_TYPE/100+3;                                    1  100
            IF ROOM_TYPE=381 THEN                                        2  101
            ROOM_CODEIN=15;                                              2  102
            ELSE                                                         2  103
               IF ROOM_TYPE=382 THEN                                     3  103
               ROOM_CODEIN=13;                                           3  104
               ELSE                                                      3  105
                  IF ROOM_TYPE=383 THEN                                  4  105
                  ROOM_CODEIN=13;                                        4  106
SEARCH: GET STRING(SUBSTR(CARDIN,20,5))EDIT(EXSPACEIN)(F(5));            1  107
            IF DEPT_CODE¬=STDEPT_CODE THEN                               2  108
               DO;                                                       3  109
/*  THIS SECTION OF THE PROGRAM PRINTS OUT THE PROCESSED INFO            */  3  110
/*  FOR THIS DEPARTMENT BEFORE COMMENCING TO PROCESS INPUT FOR THE-      */  3  110
/*  NEXT DEPARTMENT                                                      */  3  110
               II=1;                                                     3  110
               CALL PRINT;                                               3  111
               SPDATA=0;                                                 3  112
               STDEPT_CODE=DEPT_CODE;                                    3  113
               PUT SKIP FILE(OUTFILE)EDIT('TOTAL',TOTAL)(COL(85),A,COL(96  3  114
               ),P'(10)Z');                                              3  114
               GRANDTOTAL=GRANDTOTAL+TOTAL;                              3  115
               TOTAL=0;                                                  3  116
REREAD:        GET FILE(SYSIN)EDIT(DEPT_CODE1,HEGIS,DEPT_NAME)(X(10),A(4)  3  117
               ,X(1),A(4),X(1),A(20));                                   3  117
                  IF DEPT_CODE¬=DEPT_CODE1 THEN                          4  118
                  GO TO REREAD;                                          4  119
/*    PRINT THE CURRENT  DEPARTMENT  HEADINGS  */                        3  120
               PUT SKIP(2)FILE(OUTFILE)EDIT(DEPT_CODE,HEGIS,DEPT_NAME)(CO  3  120
               L(31),A,COL(37),A,COL(43),A);                             3  120
               ROOM_CODE(1)=ROOM_CODEIN;                                 3  121
               FUNCTION(1)=FUNCTIONIN;                                   3  122
               EXSPACE(1)=EXSPACEIN;                                     3  123
               GO TO SETUP;                                              3  124
               END;                                                      3  125
            DO KK=1 TO 15;                                               2  126
               IF ROOM_CODE(KK)=0 THEN                                   3  127
                  DO;                                                    4  128
                  ROOM_CODE(KK)=ROOM_CODEIN;                             4  129
                  FUNCTION(KK)=FUNCTIONIN;                               4  130
                  EXSPACE(KK)=EXSPACEIN;                                 4  131
                  GO TO SETUP;                                           4  132
                  END;                                                   4  133
               IF ROOM_CODE(KK)=ROOM_CODEIN THEN                         3  134
                  IF FUNCTION(KK)=FUNCTIONIN THEN                        4  135
                     DO;                                                 5  136
                     EXSPACE(KK)=EXSPACE(KK)+EXSPACEIN;                  5  137
                     GO TO SETUP;                                        5  138
                     END;                                                5  139
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

          END;                                                          2  140
TOTAL:    PROCEDURE;                                                    2  141
MLOOP:    INDICATOR=12;                                                 2  142
          NCARD=NCARD+1;                                                2  143
          PUT FILE(PUNCH)EDIT(REGIS,STUDEPT_CODE,NCARD)(COL(2),A,A,F(1))2  144
          ;                                                             2  144
SLOOP:       DO WHILE(ROOM_CODE(II)¬=0);                                3  145
                IF INDICATOR=32 THEN                                    4  146
                GO TO MLOOP;                                            4  147
             PUT SKIP FILE(OUTFILE)EDIT(DESCRIPTION(ROOM_CODE(II)),ROOM 3  148
             _DEF(ROOM_CODE(II)),FUNCTION(II),EXSPACE(II))(COL(65),A,CO 3  148
             L(87),A,COL(93),F(2),COL(96),P'(10)Z');                    3  148
             PUT FILE(PUNCH)EDIT(SPDATA(II))(COL(INDICATOR),F(2),F(2),F 3  149
             (5));                                                      3  149
             TOTAL=TOTAL+EXSPACE(II);                                   3  150
             INDICATOR=INDICATOR+10;                                    3  151
             II=II+1;                                                   3  152
             END;                                                       3  153
          END TOTAL;                                                    2  154
END:   END PROCESS;                                                     1  155
```

JOB CONTROL REQUIRED.

This is the necessary JCL required to sort the existing space inventory deck maintained by the university and then process it using the program PROCESS.

```
//          JOB (FH08R273,7,4),SHEKAR,CLASS=B
//SORT1  EXEC PGM=IERRC000
//SYSOUT DD SYSOUT=A
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SORTMODS DD UNIT=SYSDA,SPACE=(TRK,(10,,3))
//SYSLMOD DD UNIT=SYSDA,SPACE=(TRK,(25,15,1))
//SYSLIN DD UNIT=SYSDA,SPACE=(80,(150,10))
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(25,15))
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(02,01),,CONTIG)
//SORTOUT DD DSNAME=&RROMS,UNIT=SYSDA,DISP=(NEW,PASS),
//        SPACE=(CYL,(09,01)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//SORT1.SYSIN DD *
 SORT FIELDS=(80,1,D,60,4,A,49,3,A,52,2,A),FORMAT=CH,                X
               SIZE=E14000
 END
/*
//SORTIN DD *,DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280)
.....  SPACE INVENTORY DECK  .....
/*                                                    (cont)
```

```
//STEP2     EXEC PL1LFCLG
//PL1L.SYSIN DD *
                  ' PROCESS' source deck
/*
//GO.SYSIN DD *
.......................... ROOM TYPE CODES ..................
.............. ROOM TYPE DESCRIPTION CARDS ..................
........ DATA CARDS CONTAINING DEPT-CODE, HEGIS, DEPT-NAME .....
/*
//GO.INPUT1 DD DSNAME=*.SORT1.SORTOUT,DISP=(OLD,DELETE)
//GO.PUNCH DD SYSOUT=B,DCB=BLKSIZE=80
//GO.OUTFILE DD SYSOUT=A
/* 360/50
```

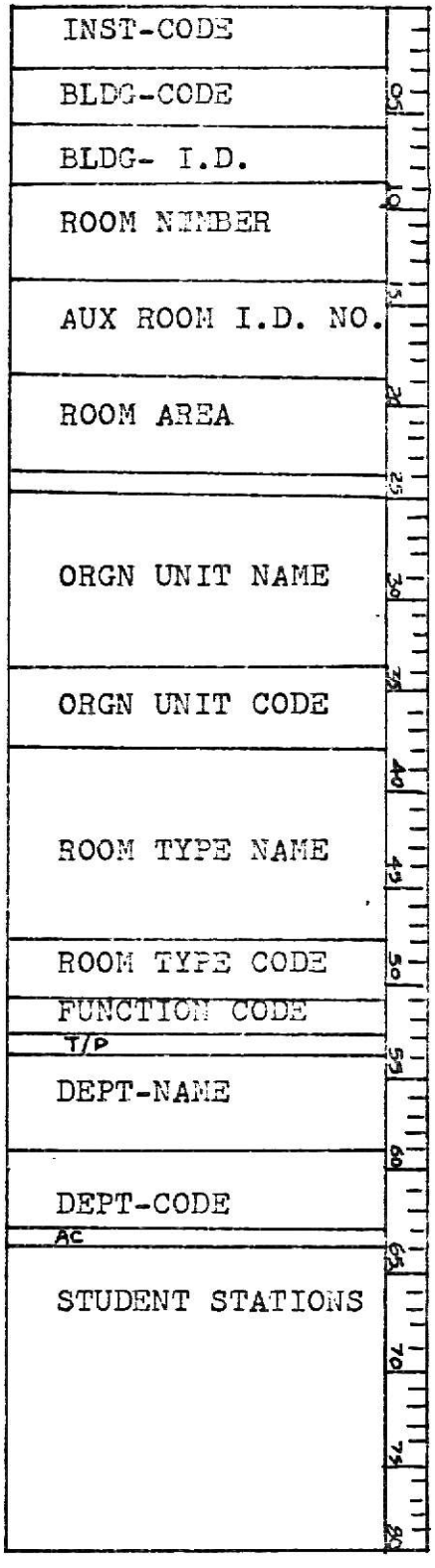


Figure A.3. Card Format of the University Space-Inventory Deck

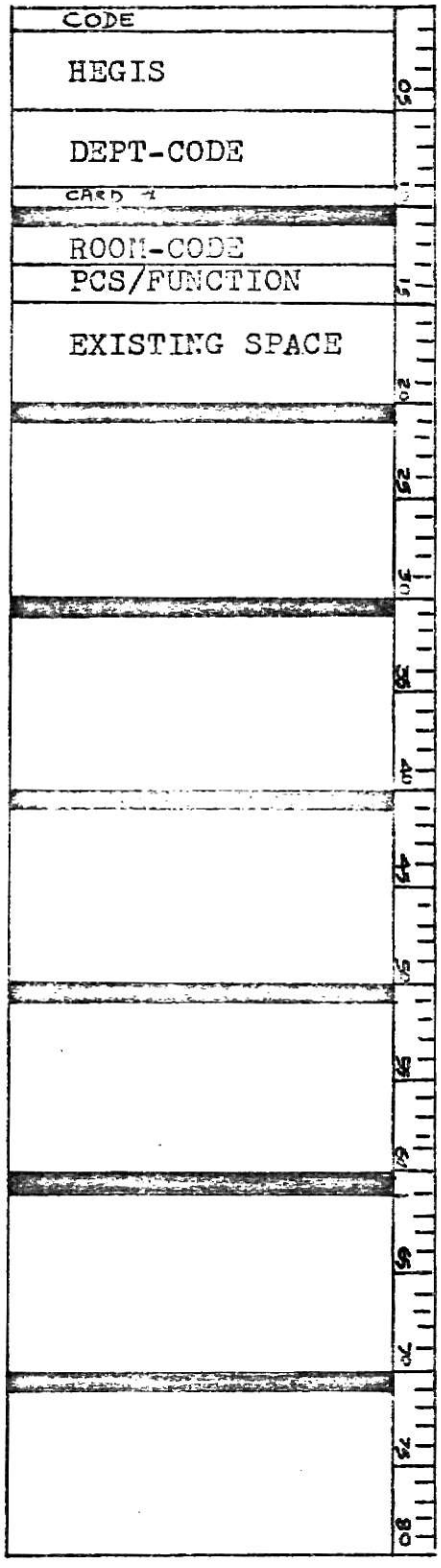


Figure A.4. Card Format for Space Inventory File-Setup and Update

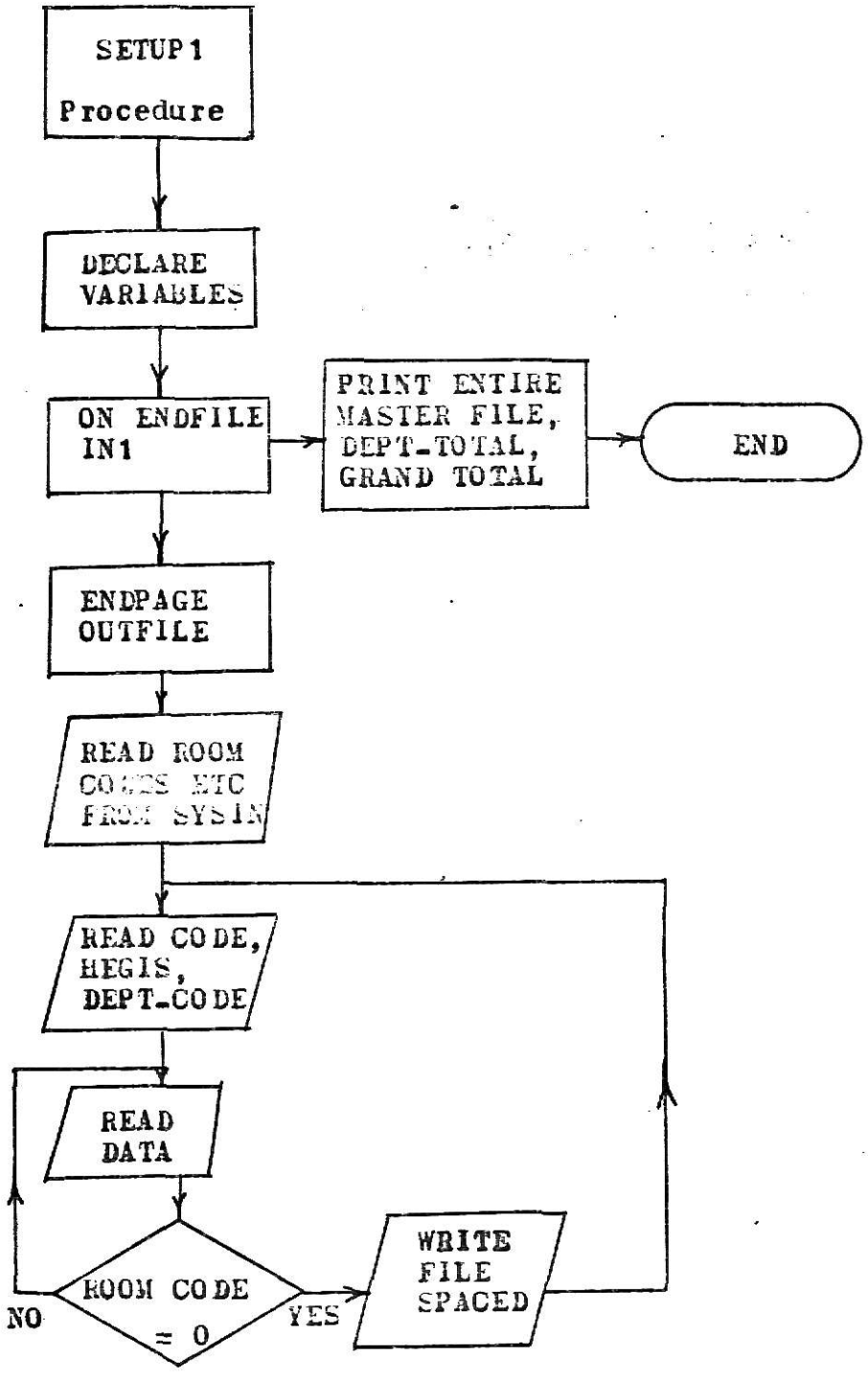


Figure A.5. Flow Chart of program SETUP1

Figure A.6. SETUP1: Source Listing and Job Control required

```
KSU'S PL/I BEATENER AND PRECOMPILER                    PAGE

SETUP1: PROCEDURE OPTIONS(MAIN);                                        1     1
/*                                                                      */  1     2
/*  ********************************************************************/  1     2
/*  THIS PROGRAM SETS UP THE EXISTING SPACE DATA FILE AS A PERMANENT */     1     2
/*  DATA SET, AND PRINTS OUT THE ENTIRE MASTER FILE                  */     1     2
/*                                                                   */     1     2
/*  ********************************************************************/  1     2
  DCL 1 SPDATA,  2  REGIS  CHAR(4), 2 ADM_CODEE  CHAR(4),                 1     2
  2  DATA(14),  3 EXROOM_CODE BIN FIXED, 3 EXPOS BIN FIXED,               1     2
  3 EXISTING_SPACE  BIN FIXED(31);                                        1     2
        DCL DEPT_NAME CHAR(20), ROOM_DEF(13) CHAR(3), DESCRIPTION(13) C   1     3
HAR(20),CODE CHAR(1), DEPT_CODE CHAR(4);                                  1     3
        DCL REGION CHAR(8) INIT ((8)' '), INTER BIN FIXED (31);           1     4
        DCL TOTAL BIN FIXED(31) INIT(0), GRANDTOTAL FIXED BIN (47) INIT   1     5
 (0);                                                                     1     5
        DCL SPACED FILE RECORD KEYED ENV (REGIONAL(3));                   1     6
        ON ENDFILE(IN2)                                                   1     7
           BEGIN;                                                         2     8
           CLOSE FILE(SPACED);                                            2     9
           OPEN FILE(SPACED)DIRECT INPUT;                                 2    10
           GO TO PRINTOUT;                                                2    11
           END;                                                           2    12
        ON ENDPAGE(OUTFILE)                                               1    13
           BEGIN;                                                         2    14
           PUT PAGE FILE(OUTFILE)EDIT('KANSAS STATE UNIVERSITY  : EXISTI  2    15
           NG SPACE SUMMARY BY ROOM TYPES AND FUNCTION')(COL(35),A);      2    15
           PUT SKIP(2)FILE(OUTFILE)EDIT('DEPT- REGIS   DEPARTMENT-NAME    2    16
                 DESCRIPTION      ROOM  POS   EXISTING')(COL(31),A)('CO   2    16
           DE';'TYPE','NASF')(COL(31),A,COL(86),A,COL(102),A);            2    16
           END;                                                           2    17
        OPEN FILE(SPACED)DIRECT OUTPUT;                                   1    18
        OPEN FILE(OUTFILE)PRINT PAGESIZE(59)LINESIZE(132);                1    19
        SIGNAL ENDPAGE(OUTFILE);                                          1    20
        GET FILE(SYSIN)EDIT((ROOM_DEF(IR)DO IR=1 TO 13))(X(1),A(3));      1    21
        GET SKIP FILE(SYSIN)EDIT(DESCRIPTION)(A(20));                     1    22
        GET SKIP FILE(SYSIN);                                             1    23
/*                                                                      */  1    24
/*                                                                      */  1    24
SET:    GET FILE(IN2)EDIT(CODE,REGIS,ADM_CODEE)(COL(1),A(1),A(4),A(4));   1    24
        DATA=0;                                                           1    25
        INDICATOR=12;                                                     1    26
        GET STRING(ADM_CODEE)EDIT(INTER)(F(4));                           1    27
           IF INTER<1499 THEN                                             2    28
           SUBSTR(REGION,8,1)='0';                                        2    29
           ELSE                                                           2    30
              IF INTER<1999 THEN                                          3    30
              SUBSTR(REGION,8,1)='1';                                     3    31
              ELSE                                                        3    32
                 IF INTER<3999 THEN                                       4    32
                 SUBSTR(REGION,8,1)='2';                                  4    33
                 ELSE                                                     4    34
                 SUBSTR(REGION,8,1)='3';                                  4    34
L2:        DO I=1 TO 14;                                                  2    35
              IF INDICATOR=82 THEN                                        3    36
              INDICATOR=12;                                               3    37
           GET FILE(IN2)EDIT(DATA(I))(COL(INDICATOR),F(2),F(2),F(5));     2    38
              IF EXROOM_CODE(I)=0 THEN                                    3    39
                 DO;                                                      4    40
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

                WRITE FILE(SPACED)FROM(SPDATA)KEYFROM(ADM_CODE||REGION      4    41
                );                                                          4    41
                GO TO SET;                                                  4    42
                END;                                                        4    43
           INDICATOR=INDICATOR+10;                                          2    44
           END L2;                                                          2    45
/*                                                                  */     1    46
/*                                                                  */     1    46
PRINTOUT:                                                                   1    46
        GET FILE(SYSIN)EDIT(DEPT_CODE,DEPT_NAME)(X(10),A(4),X(6),A(20));    1    46
           IF DEPT_CODE='0000' THEN                                         2    47
           GO TO PRINTOUT;                                                  2    48
           IF DEPT_CODE='    ' THEN                                         2    49
           GO TO OVER;                                                      2    50
        GET STRING(DEPT_CODE)EDIT(INTER)(F(4));                             1    51
           IF INTER<1499 THEN                                               2    52
           SUBSTR(REGION,8,1)='0';                                          2    53
           ELSE                                                             2    54
              IF INTER<1999 THEN                                            3    54
              SUBSTR(REGION,8,1)='1';                                       3    55
              ELSE                                                          3    56
                 IF INTER<3999 THEN                                         4    56
                 SUBSTR(REGION,8,1)='2';                                    4    57
                 ELSE                                                       4    58
                 SUBSTR(REGION,8,1)='3';                                    4    58
        PUT SKIP(2)FILE(OUTFILE)EDIT(DEPT_CODE,REGIS,DEPT_NAME)(COL(31),    1    59
        A,COL(37),A,COL(43),A);                                             1    59
PLOOP:     DO I=1 TO 14;                                                    2    60
              IF EXROOM_CODE(I)=0 THEN                                      3    61
              GOTO OUT;                                                     3    62
           PUT SKIP FILE(OUTFILE)EDIT(DESCRIPTION(EXROOM_CODE(I)),ROOM_D    2    63
           EF(EXROOM_CODE(I)),EXPOS(I),EXISTING_SPACE(I))(COL(65),A,COL(    2    63
           87),A,COL(93),F(2),COL(96),P'(10)Z');                            2    63
           TOTAL=TOTAL+EXISTING_SPACE(I);                                   2    64
           END;                                                             2    65
OUT:    PUT SKIP(2)FILE(OUTFILE)EDIT('TOTAL',TOTAL)(COL(85),A,COL(96),P'    1    66
        (10)Z');                                                            1    66
        GRANDTOTAL=GRANDTOTAL+TOTAL;                                        1    67
        TOTAL=0;                                                            1    68
        GO TO PRINTOUT;                                                     1    69
/*                                                                  */     1    70
/*                                                                  */     1    70
/*                                                                  */     1    70
/*                                                                  */     1    70
OVER:   PUT SKIP(4)FILE(OUTFILE)EDIT('GRAND-TOTAL',GRANDTOTAL)(COL(75),A    1    70
        ,COL(94),P'(12)Z')('NORMAL TERMINATION OF THE PROGRAM')(SKIP(4),    1    70
        COL(5),A);                                                          1    70
        END SETUP1;                                                         1    71
```

JOB CONTROL REQUIRED.

```
//          JOB (FH08R273,3,2),SHEKAR,CLASS=A
//          EXEC PL1LFCLG
//PL1L.SYSIN DD *
                    ' SETUP1' source deck
/*
//GO.SYSIN DD *
..................... ROOM TYPE DESCRIPTION CARDS .............
..................... ROOM TYPE CODES .....................
....... DATA CARDS CONTAINING DEPT-CODE AND DEPT-NAME
/*
//GO.IN2 DD *
.................. PROCESSED SPACE INVENTORY DECK .............
/*
//GO.SPACED DD DSN=COFH08.SPACEX,UNIT=2314,VOL=SER=111111,
//          SPACE=(120,150),DISP=(NEW,KEEP),
//          DCB=(RECFM=F,BLKSIZE=120,DSORG=DA,KEYLEN=4)
//GO.OUTFILE DD SYSOUT=A
/* 360/50
```

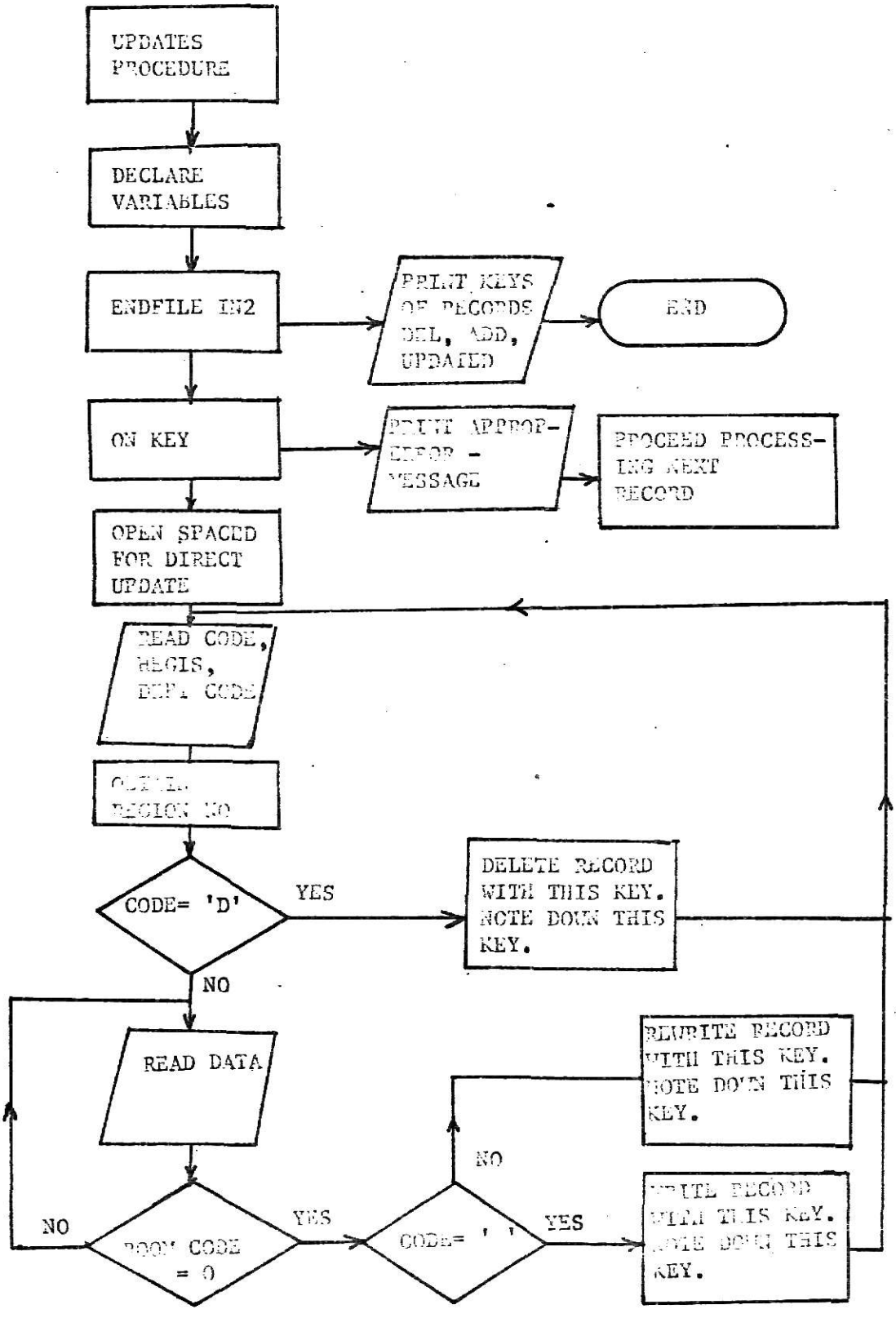


Figure A.7. Flow Chart of Program UPDATES

Figure A.8. UPDATES: Source Listings and Job Control required

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

UPDATES:PROCEDURE OPTIONS(MAIN);                                    1    1
/*                                                                  */  1    2
/*  ****************************************************************/  1    2
/*                                                                  */  1    2
/*  THIS PROGRAM WAS WRITTEN FOR FUTURE UPDATES OF THE EXISTING     */  1    2
/*  SPACE DATA FILE. THIS PROGRAM CAN BE USED TO ADD NEW RECORDS,   */  1    2
/*  UPDATE EXISTING RECORDS, AND DELETE UNWANTED RECORDS.           */  1    2
/*  THE PROGRAM LETS YOU KNOW THE RECORDS THAT WERE ADDED, UPDATED  */  1    2
/*  OR DELETED                                                      */  1    2
/*                                                                  */  1    2
/*  ****************************************************************/  1    2
                                                                    D C  1    2
L 1 SPDATA,  2  REGIS  CHAR(4), 2 ADM_CODEE  CHAR(4),               2    1    2
 DATA(14),  3 EXROOM_CODE BIN FIXED, 3 EXPOS BIN FIXED,             3    1    2
EXISTING_SPACE  BIN FIXED(31);                                           1    2
        DCL (ADD(20),DEL(20),UPDAT(20)) CHAR(4) ;                        1    3
        DCL REGION CHAR(8) INIT ((8)' '), INTER BIN FIXED (31), CODE CH  1    4
AR(1);                                                                   1    4
        DCL SPACED FILE RECORD KEYED ENV (REGIONAL(3));                  1    5
        ON ENDFILE(IN2)                                                  1    6
        GO TO PRINTOUT;                                                  1    7
        ON KEY(SPACED)                                                   1    8
           BEGIN;                                                        2    9
              IF CODE=' ' THEN                                           3   10
                 DO;                                                     4   11
                 PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODEE,'  IS AL  4   12
                 READY EXISTINGIN THE FILE. INPUT IGNORED.')(COL(5),A,A,  4   12
                 A);                                                     4   12
                 END;                                                    4   13
              ELSE                                                       3   14
                 IF CODE='U' THEN                                        4   14
                    DO;                                                  5   15
                    PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODEE,'  IS 5   16
                     NOT IN THE FILE. UPDATE IS IGNORED')(COL(5),A,A,A); 5   16
                    END;                                                 5   17
                 ELSE                                                    4   18
                    DO;                                                  5   18
                    PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODEE,'  IS 5   19
                     NOT IN THE FILE. DELETE IS IGNORED')(COL(5),A,A,A); 5   19
                    END;                                                 5   20
           GO TO SET;                                                    2   21
           END;                                                          2   22
        OPEN FILE(SPACED)DIRECT UPDATE;                                  1   23
        IA,ID,IU=1;                                                      1   24
        ADD,DEL,UPDAT=' ';                                               1   25
SET:    GET FILE(IN2)EDIT(CODE,REGIS,ADM_CODEE)(COL(1),A(1),A(4),A(4));  1   26
        DATA=0;                                                          1   27
        I=1;                                                             1   28
        INDICATOR=12;                                                    1   29
        GET STRING(ADM_CODEE)EDIT(INTER)(F(4));                          1   30
           IF INTER<1499 THEN                                            2   31
           SUBSTR(REGION,8,1)='0';                                       2   32
           ELSE                                                          2   33
              IF INTER<1999 THEN                                         3   33
              SUBSTR(REGION,8,1)='1';                                    3   34
              ELSE                                                       3   35
                 IF INTER<3999 THEN                                      4   35
                 SUBSTR(REGION,8,1)='2';                                 4   36
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

                  ELSE                                                    4   37
                  SUBSTR(REGION,3,1)='3';                                 4   37
               IF CODE='D' THEN                                           2   38
                  DO;                                                     3   39
                  DELETE FILE(SPACED)KEY(ADM_CODEE||REGION);              3   40
                  DEL(ID)=ADM_CODEE;                                      3   41
                  ID=ID+1;                                                3   42
                  GO TO SET;                                              3   43
                  END;                                                    3   44
               DO WHILE('1'B);                                            2   45
                  IF INDICATOR=82 THEN                                    3   46
                  INDICATOR=12;                                           3   47
               GET FILE(IN2)EDIT(DATA(I))(COL(INDICATOR),F(2),F(2),F(5));  2   48
                  IF EXROOM_CODE(I)=0 THEN                                3   49
                     DO;                                                  4   50
                        IF CODE=' ' THEN                                  5   51
                           DO;                                            6   52
                           WRITE FILE(SPACED)FROM(SPDATA)KEYFROM(ADM_CODEE||  6   53
                           REGION);                                       6   53
                           ADD(IA)=ADM_CODEE;                             6   54
                           IA=IA+1;                                       6   55
                           GO TO SET;                                     6   56
                           END;                                           6   57
                        ELSE                                              5   58
                           DO;                                            6   58
                           REWRITE FILE(SPACED)FROM(SPDATA)KEY(ADM_CODEE||RE  6   59
                           GION);                                         6   59
                           UPDAT(IU)=ADM_CODEE;                           6   60
                           IU=IU+1;                                       6   61
                           GO TO SET;                                     6   62
                           END;                                           6   63
                     END;                                                 4   64
                  IF I>=1 THEN                                            3   65
                     IF EXROOM_CODE(I)=EXROOM_CODE(I-1)THEN               4   66
                        IF EXPCS(I)=EXPCS(I-1)THEN                        5   67
                           DO;                                            6   68
                           EXISTING_SPACE(I-1)=EXISTING_SPACE(I-1)+EXISTING_  6   69
                           SPACE(I);                                      6   69
                           DATA(I)=0;                                     6   70
                           GO TO LOOP;                                    6   71
                           END;                                           6   72
               I=I+1;                                                     2   73
LOOP:          INDICATOR=INDICATOR+10;                                    2   74
               END;                                                       2   75
PRINTOUT:                                                                 1   76
            PUT SKIP(4)EDIT('THE RECORDS WITH FOLLOWING KEYS WERE ADDED TO T  1   76
            HE FILE : ')(A);                                              1   76
               IF IA=1 THEN                                               2   77
               PUT SKIP EDIT('NONE')(COL(5),A);                           2   78
               ELSE                                                       2   79
               PUT SKIP EDIT(ADD)(COL(5),(20)(X(4),A));                   2   79
            PUT SKIP(4)EDIT('THE RECORDS WITH THE FOLLOWING KEYS WERE UPDATE  1   80
            D :')(A);                                                     1   80
               IF IU=1 THEN                                               2   81
               PUT SKIP EDIT('NONE')(COL(5),A);                           2   82
               ELSE                                                       2   83
               PUT SKIP EDIT(UPDAT)(COL(5),(20)(X(4),A));                 2   83
            PUT SKIP(4)EDIT('THE RECORDS WITH THE FOLLOWING KEYS WERE DELETE  1   84
```

```
        D :')(A);                                                    1   84
           IF ID=1 THEN                                              2   85
           PUT SKIP EDIT('NONE')(COL(5),A);                          2   86
           ELSE                                                      2   87
           PUT SKIP EDIT(DEL)(COL(5),(20)(X(4),A));                  2   87
        END UPDATES;                                                 1   88
```

JOB CONTROL REQUIRED.

```
//          JOB (FH08R273,3,2),SHEKAR,CLASS=A
//          EXEC PL1LFCLG
//PL1L.SYSIN DD *
                  ' UPDATES' source deck
/*
//GO.IN2 DD *
.................. SAMPLE DATA CARDS ..................
U 99005911 127179002 1236 1587 1241 3366 1263 5223 1281 5834
/*
//GO.SPACED DD DSN=COFH08.SPACEX,UNIT=2314,VOL=SER=111111,
//          SPACE=(120,150),DISP=(OLD,KEEP),
//          DCB=(RECFM=F,BLKSIZE=120,DSORG=DA,KEYLEN=4)
/* 360/50
```

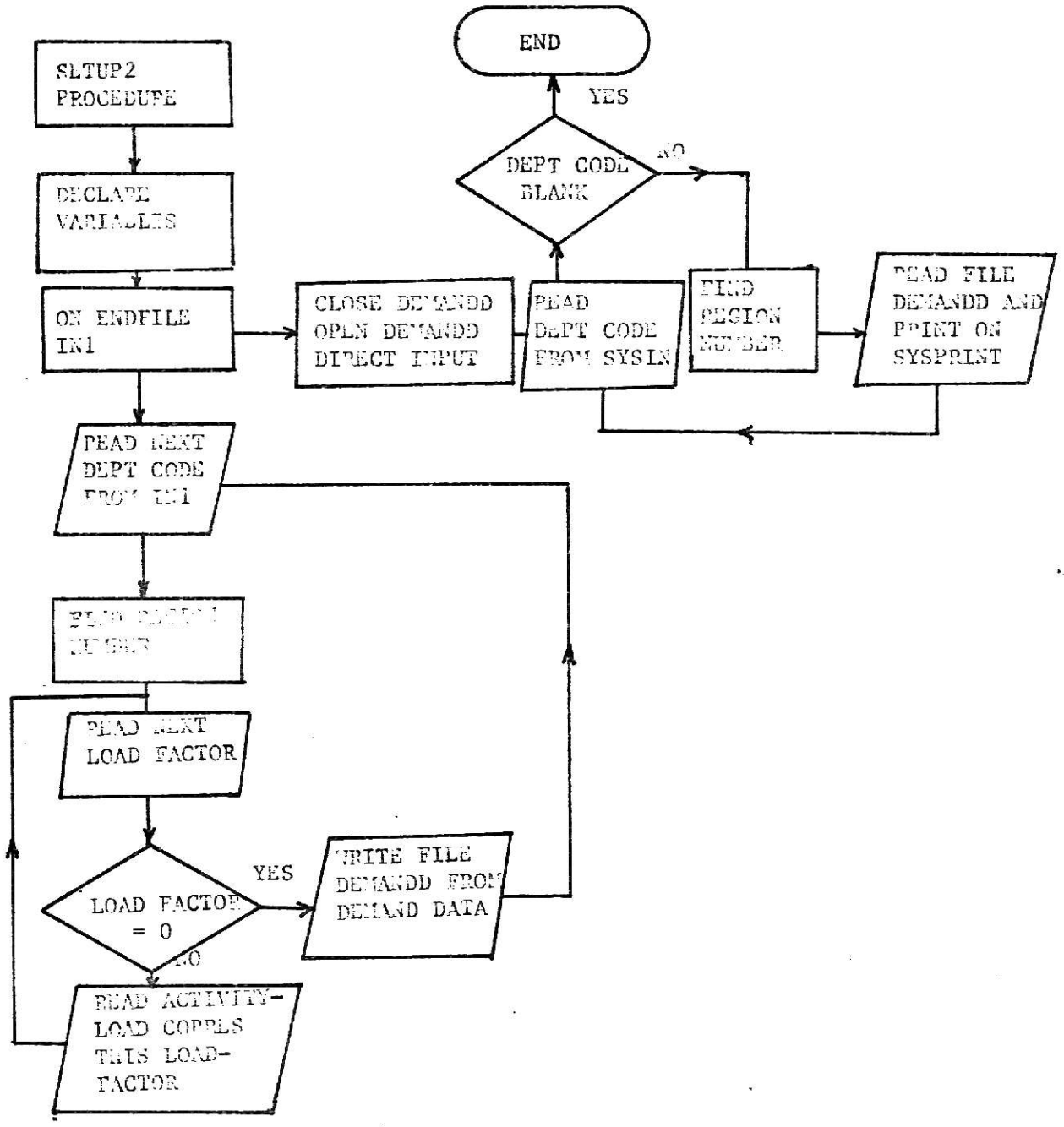


Figure A.9. Flow Chart of Program SETUP2

Figure A.10. SETUP2: Source Listings and Job Control required

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

SETUP2: PROCEDURE OPTIONS(MAIN);                                          1    1
/*                                                                  */    1    2
/*  ****************************************************************/    1    2
/*  THIS PROGRAM SETS UP THE ACTIVITY LOAD DATA FILE AS A PERMANENT  */    1    2
/*  DATA SET, AND PRINTS OUT A FEW RECORDS AS A ECHO CHECK           */    1    2
/*                                                                  */    1    2
/*  ****************************************************************/    1    2
  DCL 1 DEMAND_DATA,  2 REGIS CHAR(4), 2 ADM_CODED CHAR(4),               1    2
  2 DEPT_NAME CHAR(20), 2 DATA_TYPE(21) FIXED DEC(5,1);                   1    2
       DCL CODE CHAR(1), DEPT_CODE CHAR(4), INTER BIN FIXED(31);          1    3
       DCL REGION CHAR(8) INIT ((8)' ');                                  1    4
       DCL DEMAND FILE RECORD KEYED ENV (REGIONAL(3));                    1    5
       ON ENDFILE(IN1)                                                    1    6
          BEGIN;                                                          2    7
          CLOSE FILE(DEMANDD);                                            2    8
          OPEN FILE(DEMANDD)DIRECT INPUT;                                 2    9
          GO TO PRINTOUT;                                                 2   10
          END;                                                            2   11
       OPEN FILE(DEMANDD)DIRECT OUTPUT;                                   1   12
SET1:  GET FILE(IN1)EDIT(CODE,DEMAND_DATA.REGIS,ADM_CODED,DEPT_NAME)(CO    1   13
       L(1),A(1),X(1),A(4),A(4),X(2),A(20));                              1   13
       DATA_TYPE=0.0;                                                     1   14
       INDICATOR=34;                                                      1   15
       GET STRING(ADM_CODED)EDIT(INTER)(F(4));                            1   16
          IF INTER<1499 THEN                                              2   17
          SUBSTR(REGION,8,1)='0';                                         2   18
          ELSE                                                            2   19
             IF INTER<1999 THEN                                           3   19
             SUBSTR(REGION,8,1)='1';                                      3   20
             ELSE                                                         3   21
                IF INTER<3999 THEN                                        4   21
                SUBSTR(REGION,8,1)='2';                                   4   22
                ELSE                                                      4   23
                SUBSTR(REGION,8,1)='3';                                   4   23
L1:       IF INDICATOR=82 THEN                                            2   24
          INDICATOR=34;                                                   2   25
       GET FILE(IN1)EDIT(LF)(COL(INDICATOR),F(2));                        1   26
          IF LF=0 THEN                                                    2   27
             DO;                                                          3   28
             WRITE FILE(DEMANDD)FROM(DEMAND_DATA)KEYFROM(ADM_CODED||REG   3   29
             ION);                                                        3   29
             GO TO SET1;                                                  3   30
             END;                                                         3   31
       GET FILE(IN1)EDIT(DATA_TYPE(LF))(F(5,1));                          1   32
       INDICATOR=INDICATOR+8;                                             1   33
       GO TO L1;                                                          1   34
PRINTOUT:                                                                 1   35
       GET FILE(SYSIN)EDIT(DEPT_CODE)(X(1),A(4));                         1   35
          IF DEPT_CODE='    ' THEN                                        2   36
          GO TO OVER;                                                     2   37
       GET STRING(DEPT_CODE)EDIT(INTER)(F(4));                            1   38
          IF INTER<1499 THEN                                              2   39
          SUBSTR(REGION,8,1)='0';                                         2   40
          ELSE                                                            2   41
             IF INTER<1999 THEN                                           3   41
             SUBSTR(REGION,8,1)='1';                                      3   42
             ELSE                                                         3   43
                IF INTER<3999 THEN                                        4   43
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

                SUBSTR(REGION,6,1)='2';                                  4   44
                ELSE                                                     4   45
                SUBSTR(REGION,6,1)='3';                                  4   45
        READ FILE(DEMAND0)INTO(DEMAND_DATA)KEY(DEPT_CODE||REGION);       1   46
        PUT SKIP FILE(SYSPRINT)EDIT(DEMAND_DATA)(A,X(1),A,X(1),A,(21)(X(  1   47
    -1),F(7,1)));                                                        1   47
        GO TO PRINTOUT;                                                  1   48
OVER:   PUT SKIP(4)FILE(SYSPRINT)EDIT('NORMAL PROGRAM TERMINATION')(COL(  1   49
        5),A);                                                           1   49
        END SETUP2;                                                      1   50
```

JOB CONTROL REQUIRED.

```
//          JOB (FH08R273,3,2),SHEKAR,CLASS=A
//          EXEC PL1LFCLG
//PL1L.SYSIN DD *
                ' SETUP2' source deck
/*
//GO.IN1 DD *
..................... ACTIVITY LOAD DATA CARDS ..............
/*
//GO.SYSIN DD *
..................... DEPARTMENT CODES .....................
/*
//GO.DEMANDD DD DSN=COFH08.DEMAND,UNIT=2314,VOL=SER=111111,
//          SPACE=(91,150),DISP=(NEW,KEEP),
//          DCB=(RECFM=F,BLKSIZE=91,DSORG=DA,KEYLEN=4)
/* 360/50
```

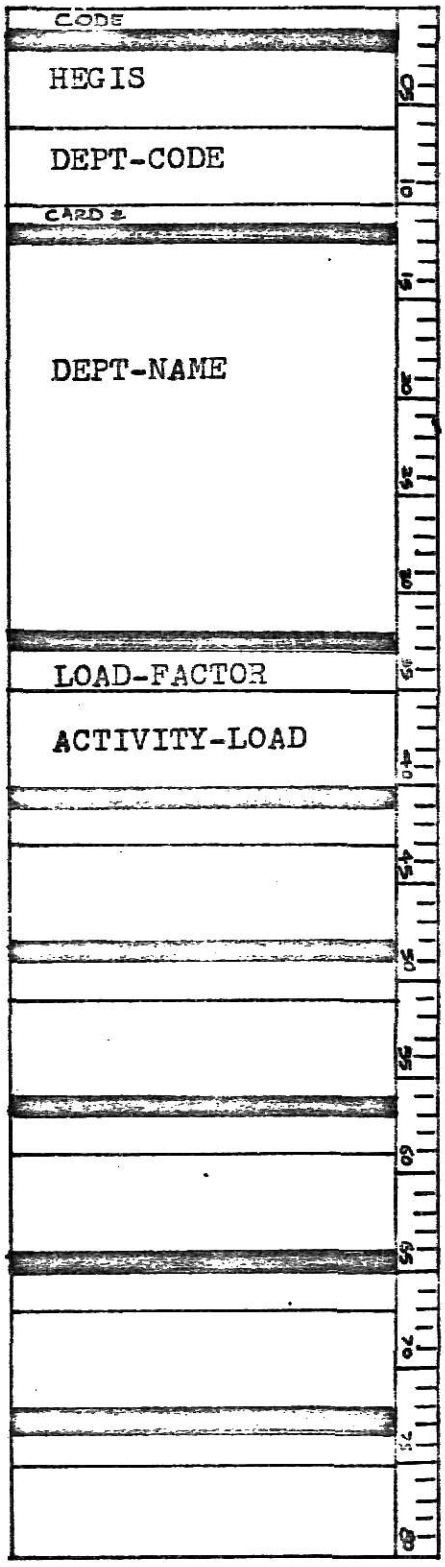


Figure A.11. Input Card Format for Demand File Setup and Update

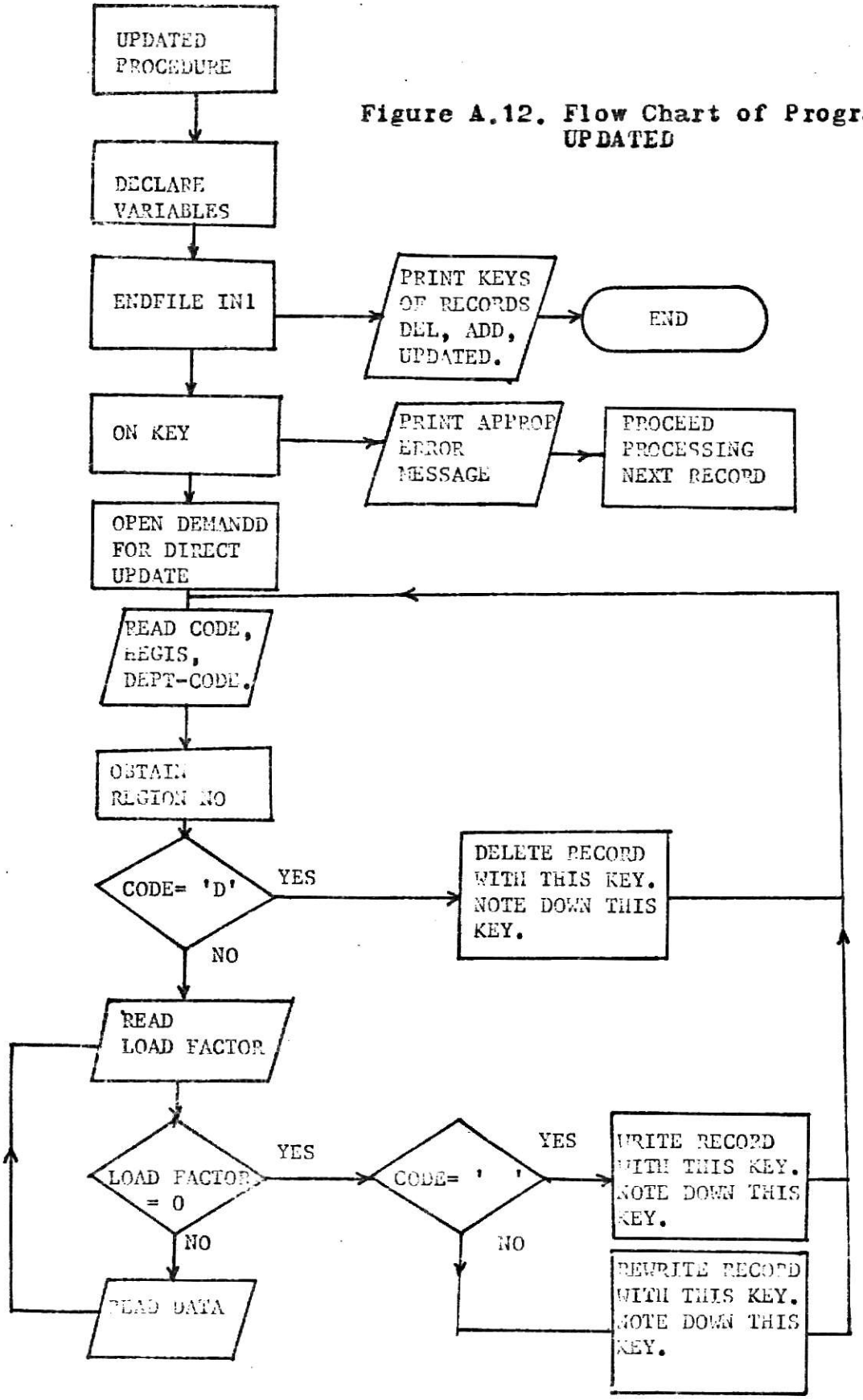


Figure A.12. Flow Chart of Program UPDATED

Figure A.13. UPDATED: Source Listings and Job Control required

```
KSU'S PL/I BEATENER AND PRECOMPILER                    PAGE

UPDATED:PROCEDURE OPTIONS(MAIN);                                        1    1
/*                                                                   */ 1    2
/*  ***************************************************************/ 1    2
/*                                                                   */ 1    2
/*  THIS PROGRAM WAS WRITTEN FOR FUTURE UPDATES OF THE ACTIVITY-     */ 1    2
/*  LOAD FILE. THIS PROGRAM CAN BE USED TO ADD NEW RECORDS,          */ 1    2
/*  UPDATE EXISTING RECORDS, AND DELETE UNWANTED RECORDS.            */ 1    2
/*  THE PROGRAM LETS YOU KNOW THE RECORDS THAT WERE ADDED, UPDATED   */ 1    2
/*  OR DELETED                                                       */ 1    2
/*                                                                   */ 1    2
/*  ***************************************************************/ 1    2
                                                                        D C  1    2
L 1 DEMAND_DATA,  2 REGIS CHAR(4), 2 ADM_CODED CHAR(4),                 2    1    2
DEPT_NAME CHAR(20), 2 DATA_TYPE(21) FIXED DEC(5,1);                          1    2
        DCL CODE CHAR(1), DEPT_CODE CHAR(4), INTER BIN FIXED(31);            1    3
        DCL REGION CHAR(8) INIT ((8)' ');                                    1    4
        DCL DEMANDD FILE RECORD KEYED ENV (REGIONAL(3));                     1    5
        DCL (ADD(20),DEL(20),UPDAT(20)) CHAR(4);                             1    6
        ON ENDFILE(IN1)                                                      1    7
        GO TO PRINTOUT;                                                      1    8
        ON KEY(DEMANDD)                                                      1    9
           BEGIN;                                                            2   10
              IF CODE=' ' THEN                                               3   11
                 DO;                                                         4   12
                 PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODED,'  IS AL     4   13
                 READY EXISTINGIN THE FILE. INPUT IGNORED.')(COL(5),A,A,     4   13
                 A);                                                         4   13
                 END;                                                        4   14
              ELSE                                                           3   15
                 IF CODE='U' THEN                                            4   15
                    DO;                                                      5   16
                    PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODED,'  IS     5   17
                     NOT IN THE FILE. UPDATE IS IGNORED')(COL(5),A,A,A);     5   17
                    END;                                                     5   18
                 ELSE                                                        4   19
                    DO;                                                      5   19
                    PUT SKIP(4)EDIT('RECORD WITH KEY = ',ADM_CODED,'  IS     5   20
                     NOT IN THE FILE. DELETE IS IGNORED')(COL(5),A,A,A);     5   20
                    END;                                                     5   21
           GO TO SET;                                                        2   22
           END;                                                              2   23
        OPEN FILE(DEMANDD)DIRECT UPDATE;                                     1   24
        IA,ID,IU=1;                                                          1   25
        ADD,DEL,UPDAT=' ';                                                   1   26
SET:    GET FILE(IN1)EDIT(CODE,DEMAND_DATA,REGIS,ADM_CODED,DEPT_NAME)(CO     1   27
        L(1),A(1),X(1),A(4),A(4),X(2),A(20));                                1   27
        DATA_TYPE=0.0;                                                       1   28
        INDICATOR=34;                                                        1   29
        GET STRING(ADM_CODED)EDIT(INTER)(F(4));                              1   30
           IF INTER<1499 THEN                                                2   31
           SUBSTR(REGION,6,1)='0';                                           2   32
           ELSE                                                              2   33
              IF INTER<1999 THEN                                             3   33
              SUBSTR(REGION,3,1)='1';                                        3   34
              ELSE                                                           3   35
                 IF INTER<3999 THEN                                          4   35
                 SUBSTR(REGION,3,1)='2';                                     4   36
                 ELSE                                                        4   37
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

                SUBSTR(REGION,2,1)='3';                                    4   37
             IF CODE='D' THEN                                              2   38
                DO;                                                        3   39
                DELETE FILE(DEMANDD)KEY(ADM_CODED||REGION);                3   40
                DEL(ID)=ADM_CODED;                                         3   41
                ID=ID+1;                                                   3   42
                GO TO SET;                                                 3   43
                END;                                                       3   44
             DO WHILE('1'B);                                               2   45
                IF INDICATOR=82 THEN                                       3   46
                INDICATOR=34;                                              3   47
             GET FILE(IN1)EDIT(LF)(COL(INDICATOR),F(2));                   2   48
                IF LF=0 THEN                                               3   49
                   DO;                                                     4   50
                      IF CODE=' ' THEN                                     5   51
                         DO;                                               6   52
                         WRITE FILE(DEMANDD)FROM(DEMAND_DATA)KEYFROM(ADM_C  6   53
                         ODED||REGION);                                    6   53
                         ADD(IA)=ADM_CODED;                                6   54
                         IA=IA+1;                                          6   55
                         GO TO SET;                                        6   56
                         END;                                              6   57
                      ELSE                                                 5   58
                         DO;                                               6   58
                         REWRITE FILE(DEMANDD)FROM(DEMAND_DATA)KEY(ADM_COD  6   59
                         ED||REGION);                                      6   59
                         UPDAT(IU)=ADM_CODED;                              6   60
                         IU=IU+1;                                          6   61
                         GO TO SET;                                        6   62
                         END;                                              6   63
                   END;                                                    4   64
             GET FILE(IN1)EDIT(DATA_TYPE(LF))(F(5,1));                     2   65
             INDICATOR=INDICATOR+8;                                        2   66
LOOP:        END;                                                          2   67
PRINTOUT:                                                                  1   68
          PUT SKIP(4)EDIT('THE RECORDS WITH FOLLOWING KEYS WERE ADDED TO T 1   68
          HE FILE : ')(A);                                                 1   68
             IF IA=1 THEN                                                  2   69
             PUT SKIP EDIT('NONE')(COL(5),A);                              2   70
             ELSE                                                          2   71
             PUT SKIP EDIT(ADD)(COL(5),(20)(X(4),A));                      2   71
          PUT SKIP(4)EDIT('THE RECORDS WITH THE FOLLOWING KEYS WERE UPDATE 1   72
          D :')(A);                                                        1   72
             IF IU=1 THEN                                                  2   73
             PUT SKIP EDIT('NONE')(COL(5),A);                              2   74
             ELSE                                                          2   75
             PUT SKIP EDIT(UPDAT)(COL(5),(20)(X(4),A));                    2   75
          PUT SKIP(4)EDIT('THE RECORDS WITH THE FOLLOWING KEYS WERE DELETE 1   76
          D :')(A);                                                        1   76
             IF ID=1 THEN                                                  2   77
             PUT SKIP EDIT('NONE')(COL(5),A);                              2   78
             ELSE                                                          2   79
             PUT SKIP EDIT(DEL)(COL(5),(20)(X(4),A));                      2   79
          END UPDATED;                                                     1   80
```

JOB CONTROL REQUIRED.

```
//          JOB (FHO8R273,3,2),SHEKAR,CLASS=A
//          EXEC PL1LFCLG
//PL1L.SYSIN DD *
                    ' UPDATED' source deck
/*
//GO.IN1 DD *
.................. SAMPLE DATA CARDS .................
U 9900 1908 CLASSROOMS              015.6
       8100 UNASSIGNABLE, ETC
/*
//GO.DEMANDD DD DSN=COFHO8.DEMAND,UNIT=2314,VOL=SER=111111,
//            SPACE=(91,150),DISP=(OLD,KEEP),
//            DCB=(RECFM=F,BLKSIZE=91,DSORG=DA,KEYLEN=4)
/* 360/50
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

SFACTOR:PROCEDURE OPTIONS(MAIN);                                    1    1
        DCL DEPT_NAME CHAR(20), HEGIS CHAR(4);                      1    2
        DCL (RUR(3),SOR(3),SOFTPERSTN(3)) FIXED DEC(6,3),           1    3
        SPACE_FACTOR(3) FIXED DEC(7,3);                             1    3
        ON ENDPAGE(SYSPRINT)                                        1    4
           BEGIN;                                                   2    5
           PUT PAGE FILE(SYSPRINT)EDIT('UNIT FLOOR CRITERIA :  CLASS LAB  2    6
           ORATORY')(COL(55),A);                                    2    6
           PUT SKIP(2)FILE(SYSPRINT)EDIT('HEGIS       DEPT-NAME        S  2    7
           OFT/    RUR    SOR  SPACE-')(COL(49),A)('STN','FACTOR')(SKIP,  2    7
           COL(79),A,COL(99),A);                                    2    7
           END;                                                     2    8
        ON ENDFILE(SYSIN)                                           1    9
        GO TO DONE;                                                 1   10
        SIGNAL ENDPAGE(SYSPRINT);                                   1   11
AGAIN:  GET FILE(SYSIN)EDIT(HEGIS,DEPT_NAME)(COL(1),A(4),X(1),A(20)); 1   12
        GET SKIP FILE(SYSIN)EDIT((SOFTPERSTN(I),RUR(I),SOR(I)DO I=1 TO 3  1   13
        ))(X(2),F(6,3),X(1),F(6,3),X(1),F(6,3));                    1   13
        SPACE_FACTOR=SOFTPERSTN/(RUR*SOR);                          1   14
        PUT SKIP(2)FILE(SYSPRINT)EDIT(HEGIS,DEPT_NAME,'L ',SOFTPERSTN(1)  1   15
        ,RUR(1),SOR(1),SPACE_FACTOR(1),'U ',SOFTPERSTN(2),RUR(2),SOR(2),  1   15
        SPACE_FACTOR(2),'S ',SOFTPERSTN(3),RUR(3),SOR(3),SPACE_FACTOR(3)  1   15
        )(COL(49),A,COL(55),A,COL(76),A,F(6,2),X(1),F(6,2),X(1),F(6,2),X  1   15
        (1),F(7,2),SKIP,COL(76),A,F(6,2),X(1),F(6,2),X(1),F(6,2),X(1),F(  1   15
        7,2),SKIP,COL(76),A,F(6,2),X(1),F(6,2),X(1),F(6,2),X(1),F(7,2));  1   15
        GO TO AGAIN;                                                1   16
DONE:   END SFACTOR;                                                1   17
```

Figure A.14. SFACTOR1: Source Listings

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

SFACTOR:PROCEDURE OPTIONS(MAIN);                                    1    1
        DCL DEMAND_UNIT (4) BIN FIXED INIT (15,3,3,15),             1    2
         DEMAND_FACTOR FIXED DEC(6,2), SPACE_FACTOR(4) FIXED DEC(6,2);  1  2
        DCL REGIS CHAR(4), DEPT_NAME CHAR(20);                      1    3
        ON ENDPAGE(SYSPRINT)                                        1    4
           BEGIN;                                                   2    5
           PUT PAGE FILE(SYSPRINT)EDIT('UNIT FLOOR CRITERIA :  NON CLASS  2  6
            LAB')(COL(46),A)('1  FTE RESEARCH FACULTY')(SKIP(3),COL(54),  2  6
           A)('2  FTE TEACHING FACULTY')(SKIP,COL(54),A)('3  HEAD COUNT   2  6
           G1')(SKIP,COL(54),A)('4  HEAD COUNT G2')(SKIP,COL(54),A)('REG  2  6
           IS     DEPARTMENT          EDU     RDF   SPACE-')(SKIP(2),COL  2  6
           (41),A)('FACTOR')(SKIP,COL(85),A);                       2    6
           END;                                                     2    7
        SIGNAL ENDPAGE(SYSPRINT);                                   1    8
AGAIN:  GET FILE(SYSIN)EDIT(REGIS,DEPT_NAME,DEMAND_FACTOR)(COL(1),A(4),X  1  9
        (1),A(20),X(1),F(3,1));                                     1    9
           IF REGIS='OVER' THEN                                     2   10
           GO TO OVER;                                              2   11
           DO I=1 TO 4;                                             2   12
           SPACE_FACTOR(I)=DEMAND_UNIT(I)*DEMAND_FACTOR;            2   13
              IF I=1 THEN                                           3   14
              PUT SKIP(2)FILE(SYSPRINT)EDIT(REGIS,DEPT_NAME,I,DEMAND_UNI  3  15
              T(I),DEMAND_FACTOR,SPACE_FACTOR(I))(COL(41),A,COL(47),A,CO  3  15
              L(68),F(1),COL(75),F(4),COL(77),F(5,1),COL(84),F(7,2));   3  15
              ELSE                                                  3   16
              PUT SKIP FILE(SYSPRINT)EDIT(I,DEMAND_UNIT(I),DEMAND_FACTOR  3  16
              ,SPACE_FACTOR(I))(COL(68),F(1),COL(75),F(4),COL(77),F(5,1)  3  16
              ,COL(84),F(7,2));                                     3   16
           END;                                                     2   17
        GO TO AGAIN;                                                1   18
OVER:   END SFACTOR;                                                1   19
```

Figure A.15. SFACTOR2: Source Listings

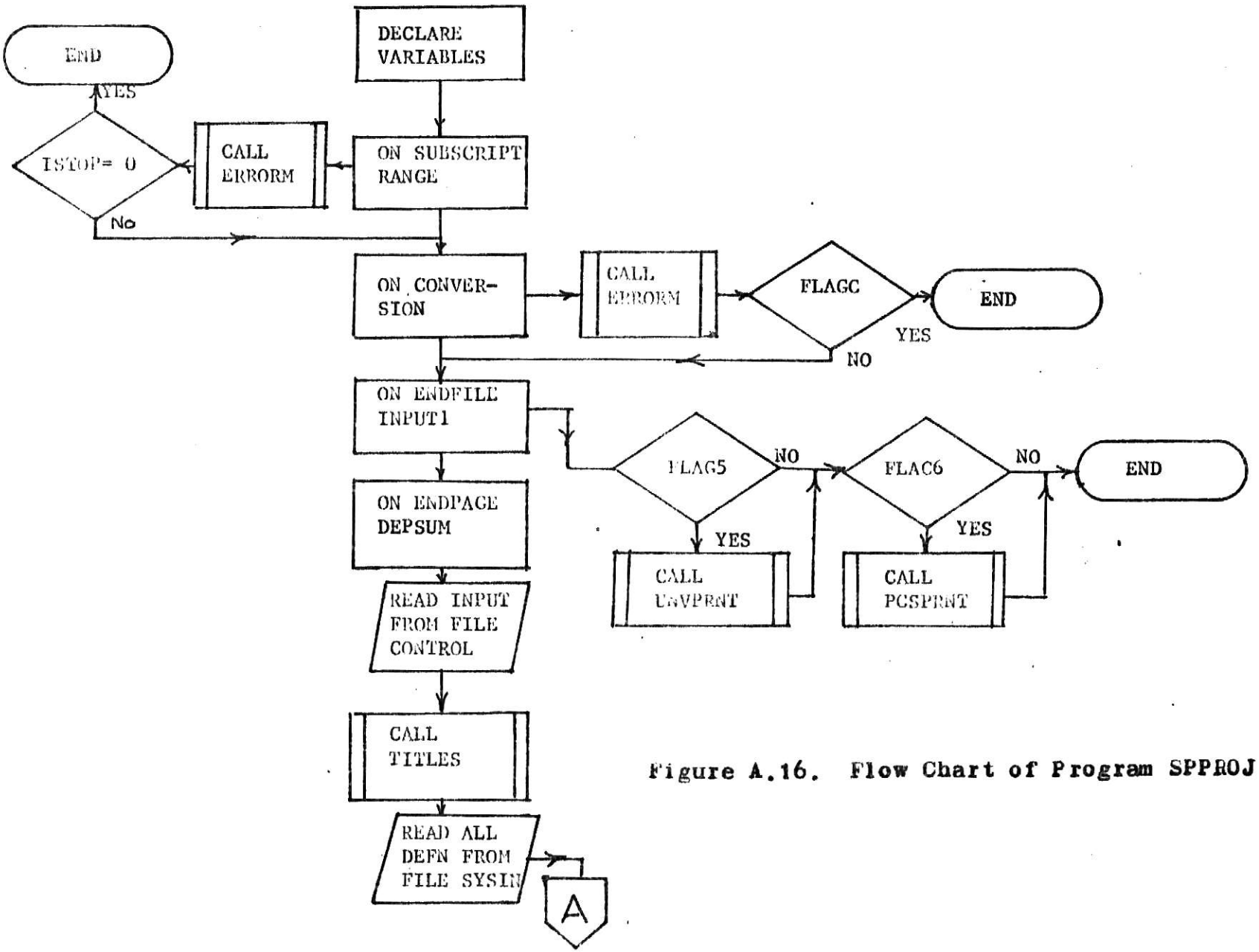


Figure A.16. Flow Chart of Program SPPROJ

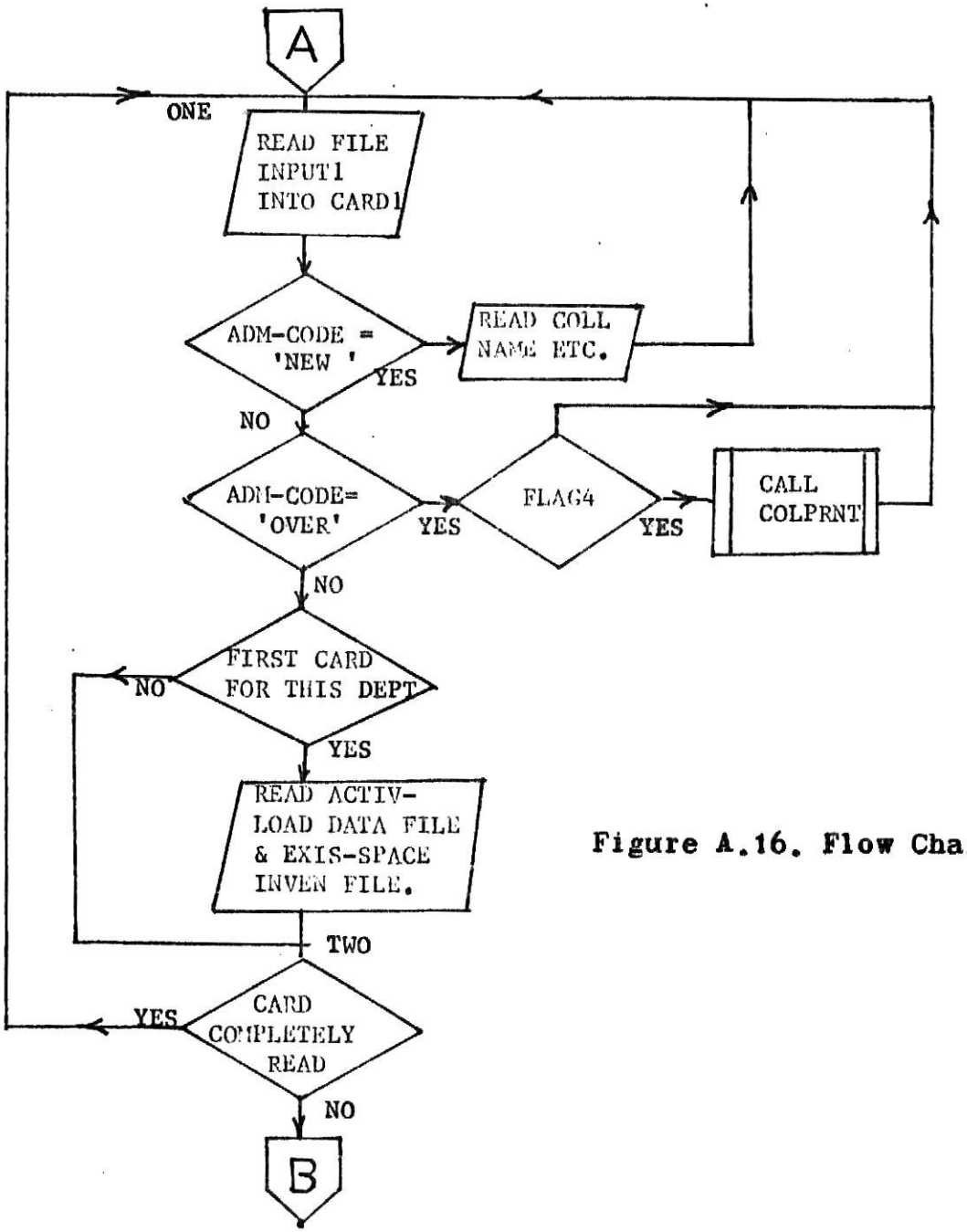


Figure A.16. Flow Chart of SPPROJ (continued)

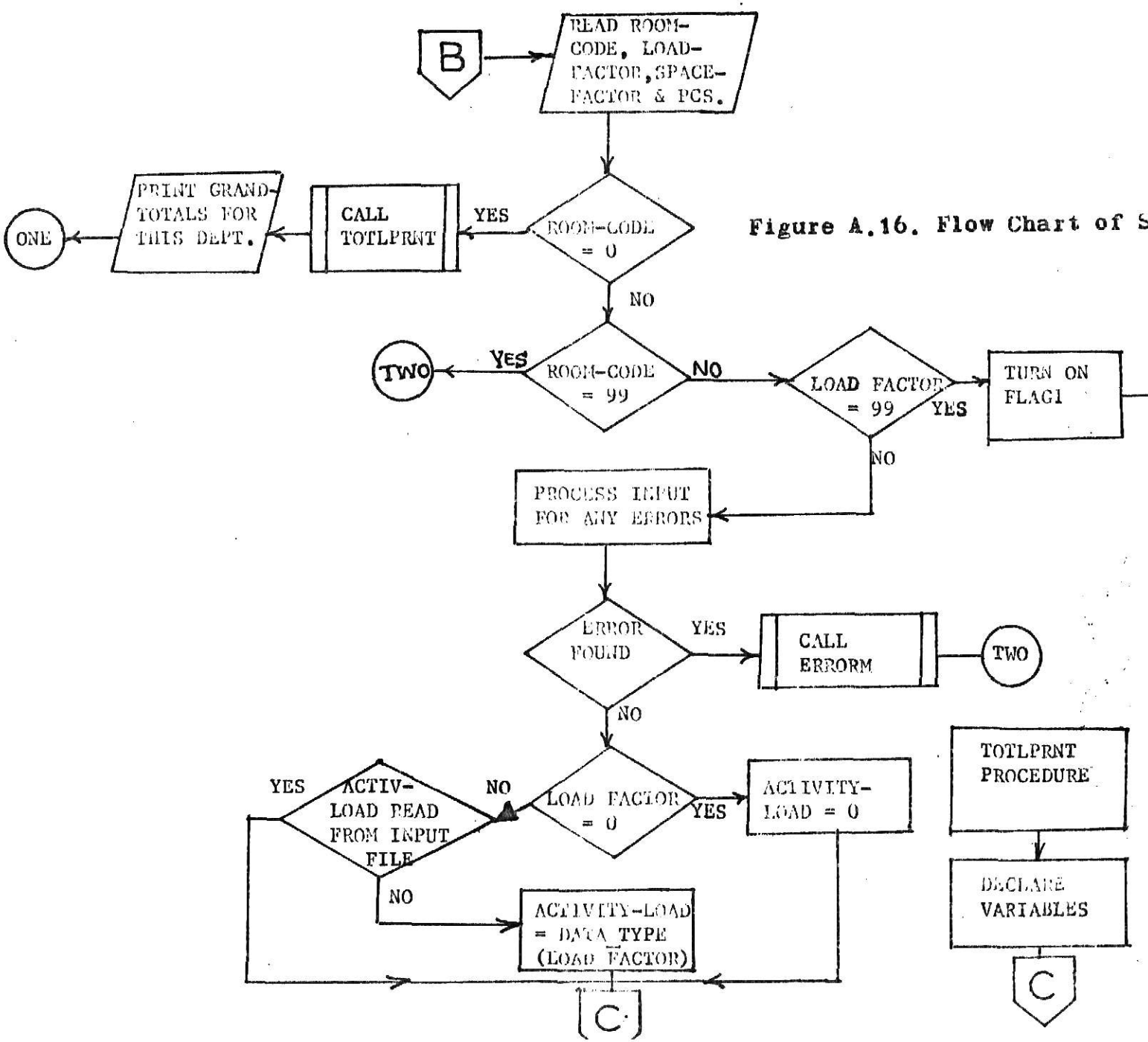


Figure A.16. Flow Chart of SPPROJ (continued)

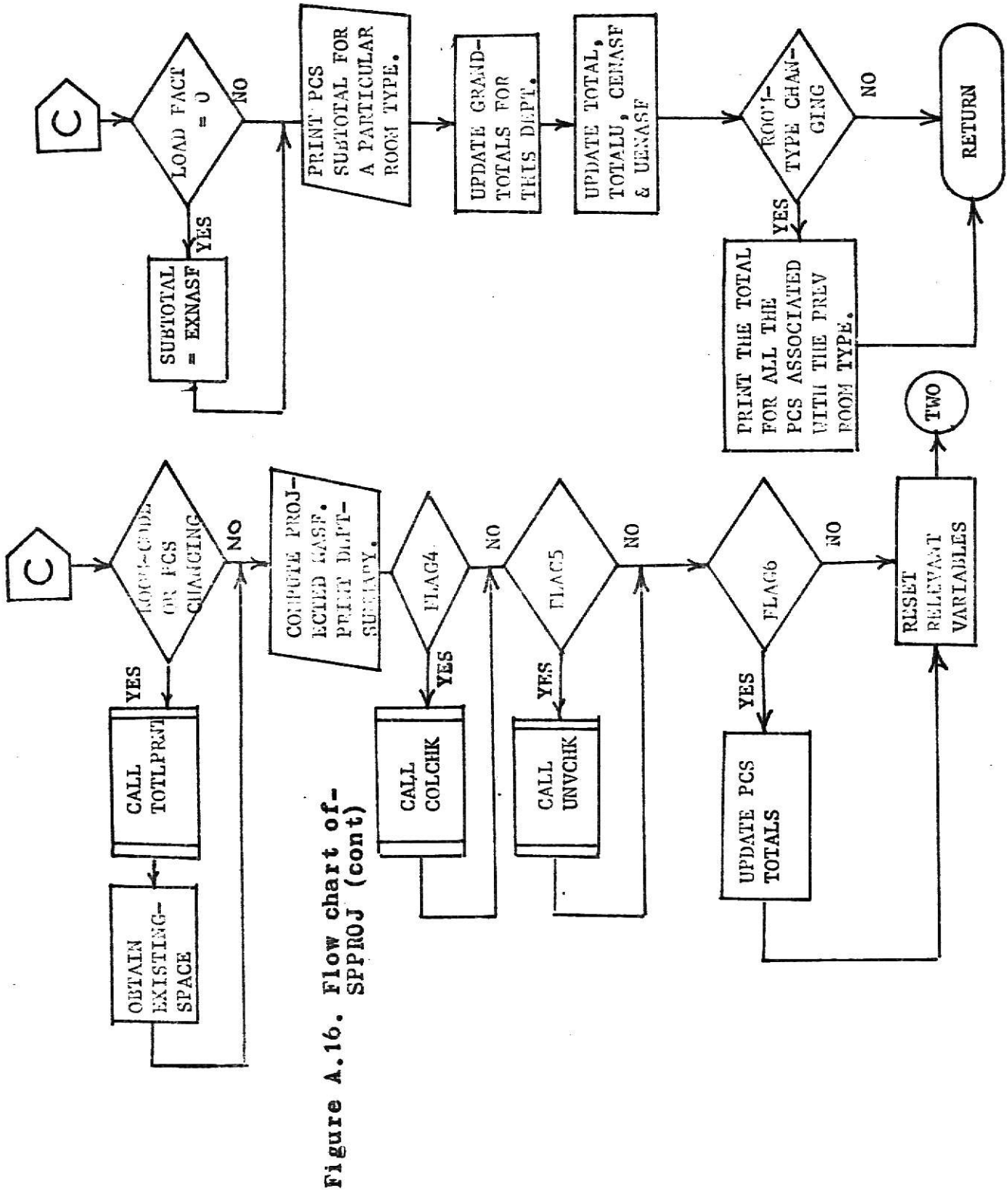


Figure A.16. Flow chart of-SPPROJ (cont)

Figure A.17. SFPROJ: Source Listings

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

(SUBSCRIPTRANGE,STRINGRANGE):SPREAD:                                    1    1
        PROCEDURE OPTIONS(MAIN);                                        1    1
/*************************************************************************/  1    2
/*                                                                       */  1    2
/*                  UNIVERSITY SPACE PLANNING     -                      */  1    2
/*                                                                       */  1    2
/*                     PROJECTIONS FOR KANSAS STATE UNIVERSITY           */  1    2
/*                                                                       */  1    2
/*                                                                       */  1    2
/*                                                                       */  1    2
/*                                                                       */  1    2
/*  ............. DEFINITIONS OF IMPORTANT VARIABLES ..................*/  1    2
/*                                                                       */  1    2
/*  STPCSC(13,10) : STORES ALL PCS FOR EACH ROOM TYPE, FOR EACH-         */  1    2
/*  COLLEGE. A MAXIMUM OF 10 PCS VALUES PER ROOM TYPE ASSUMED            */  1    2
/*  STPCSU(13,32) : STORES ALL THE PCS FOR EACH ROOM TYPE, FOR THE       */  1    2
/*  ENTIRE UNIVERSITY. A MAXIMUM OF 32 VALUES PER ROOM TYPE ASSUMED      */  1    2
/*  CALOAD(13,10,5): FOR A PARTICULAR ROOM TYPE & PCS THIS ARRAY-        */  1    2
/*  ACCUMULATES THE ACTIVITY LOAD DATA, FOR THE DIFFERENT LOAD-          */  1    2
/*  FACTORS POSSIBLE FOR THIS ROOM TYPE  ( FOR COLLEGE )                 */  1    2
/*  UALOAD(13,32,5) : SAME AS ABOVE  (FOR THE ENTIRE UNIVERSITY)         */  1    2
/*  CPNASF(13,10,5) : ACCUMULATES PROJECTED NASF FOR EACH ROOM TYPE      */  1    2
/*  BY PCS AND LOAD FACTORS BREAKDOWNS (FOR COLLEGE )                    */  1    2
/*  UPNASF(13,32,5) : SAME AS ABOVE  (FOR THE ENTIRE UNIVERSITY )        */  1    2
/*  TOTAL(13,10) : ACCUMULATES PROJECTED NASF FOR EACH ROOM TYPE AND     */  1    2
/*  ASSOCIATED PCS  (FOR COLLEGE)                                        */  1    2
/*  TOTALU(13,10) : ACCUMULATES PROJECTED NASF FOR EACH ROOM TYPE        */  1    2
/*  AND ASSOCIATED PCS  (FOR UNIVERSITY )                                */  1    2
/*  NR..... MAXIMUM NUMBER OF ROOM TYPES  (13)                           */  1    2
/*  NC..... MAXIMUM NUMBER OF PCS PER ROOM TYPE (FOR COLLEGE);           */  1    2
/*  TEN VALUES ASSUMED                                                   */  1    2
/*  NU..... SAME AS ABOVE (FOR UNIVERSITY); 32 VALUES ASSUMED            */  1    2
/*  NL..... MAXIMUM SUBCATEGORIES FOR EACH ROOM TYPE; 5 ASSUMED          */  1    2
/*  TABLE3(13,5) : THIS MATRIX STORES THE LOAD FACTORS ASSOCIATED        */  1    2
/*  WITH EACH ROOM TYPE                                                  */  1    2
/*  UNIT(0:30) :  STORES THE UNITS OF ALL LOAD FACTORS                   */  1    2
/*  ROOMSUB(0:30) :  STORES THE DEFINITIONS OF DIFFERENT LOAD FACTORS    */  1    2
/*  PPNASF(40) : ACCUMULATES THE PROJECTED NASF CORRESPONDING TO         */  1    2
/*  DIFFERENT PCS                                                        */  1    2
/*  PENASF(40) : ACCUMULATES EXISTING NASF CORRESPONDING TO DIFFERENT    */  1    2
/*  PCS                                                                  */  1    2
/*  FLAG1, FLAG1A : THESE ARE SWITCHES WHICH ENABLE PROPER READING       */  1    2
/*  OF ACTIVITY LOAD DATA FROM THE INPUT FILE                            */  1    2
/*  FLAG2, FLAG3, SWITCH1, SWITCH2 : THESE SWITCHES CONTROL OUTPUT-      */  1    2
/*  PRINTOUT                                                             */  1    2
/*  FLAG5, FLAG6, FLAG7 : THESE SWITCHES INDICATE WHETHER COLLEGE-       */  1    2
/*  SUMMARY, UNIVERSITY-SUMMARY, AND PCS-SUMMARY ARE REQUIRED. IF        */  1    2
/*  THEY ARE ON,  THEN THE RESPECTIVE SUMMARIES ARE REQUIRED             */  1    2
/*  CARD1 :  BASED, CHAR(80) VARIABLE. INFORMATION FROM FILE INPUT1      */  1    2
/*  IS READ INTO THIS VARIABLE                                           */  1    2
/*                                                                       */  1    2
/*  *********************************************************************/  1    2
/*  ********************** DECLARES ********************************/      1    2
/*                                                                       */  1    2
    DCL 1  DEPAND_DATA,  2  REGIS CHAR(4),  2  ROM_CODED  CHAR(4),       1    2
    2  DEPT_NAME  CHAR(20),  2 DATA_TYPE (21)  FIXED DEC (5,1),          1    2
    1  EXSPDATA,  2  REGIS  CHAR(4),   2  ATP_CODEE  CHAR (4), 2        1    2
    SPDATA (14),   3 EXROOM_CODE  FIXED BIN, 3 EXPCS  FIXED BIN,        1    2
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

     3  EXISTING_SPACE  FIXED BIN (31);                                    1    2
        DCL STPCSC(13,10) BIN FIXED, CPNASF(13,10) BIN FIXED(31),          1    3
        CALOAD(13,10,35) FIXED DEC(10,2), CPMASF(13,10,35) BIN FIXED(31    1    3
),      TOTAL(13,10) BIN FIXED(31);                                        1    3
        DCL STPCSU(13,30) BIN FIXED,  UPNASF(13,30) BIN FIXED(31),         1    4
        UALOAD(13,30,35) FIXED DEC(10,2), UPMASF(13,30,35) BIN FIXED(31    1    4
),      TOTALU(13,30) BIN FIXED(31);                                       1    4
        DCL TABLE3(13,35) BIN FIXED,  ROOM (13) CHAR(3), DESCRIP(13) CH    1    5
AR(20), LENGTH(13) BIN FIXED, LENUNIV(13) BIN FIXED;                       1    5
        DCL (NR,NC,NU,NL) EXTERNAL ;                                       1    6
        DCL (FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG6) BIT(1) INIT ('1'B),      1    7
        FLAG2A BIT(1) INIT ('0'B),  FLAG1A BIT (1) INIT ('0'B), FLAGC B    1    7
IT(1),      CARD1 CHAR(80) BASED(P),                                       1    7
        (CARDCOUNT1,CARD#1,TOTAL_CARDS1) BIN FIXED, STADM_CODE CHAR(4)     1    7
INIT    ('    '),                                                          1    7
            ACTIVITY_LOAD FIXED DEC (7,2) INIT (0.0),                      1    7
             (ROOM_CODE,STROOM_CODE) BIN FIXED,                            1    7
                        PROJNASF BIN FIXED (31), SPACE_FACTOR FIXED DE     1    7
C (6,2),EXNASF BIN FIXED (31), (SWITCH1,SWITCH2) BIT(1),                   1    7
            TOTAL1 BIN FIXED(31) INIT (0),   DATA1  CHAR(11),              1    7
             CODE CHAR(1) , (PCS,STPCSCODE) BIN FIXED,                     1    7
            (GRANDTOTAL1,GRANDTOTAL2) BIN FIXED (31) INIT (0),             1    7
            (COL_SUMMARY,UNIV_SUMMARY,PCS_SUMMARY) CHAR(4),                1    7
            INTEP BIN FIXED (31), REGION CHAR(8) INIT ((8)' '),            1    7
            PCSSWITCH (8) BIT(1) INIT ((8) '0'B),  ADM_CODE CHAR(4) INI    1    7
T (' ');                                                                   1    7
        DCL ( ROOMSUB (0:3))    CHAR(3), UNIT (0:30) CHAR(3),              1    8
                (ADMC,REGISC)      CHAR(4) INIT ('    '), PCSDEF(3) CH    1    8
AR(20), COLLEGE      CHAR(20)) EXTERNAL;                                   1    8
        DCL ( PCSVAL(40)  FIXED BIN, PRNASF  (40)BIN FIXED(31),            1    9
          PENASF(40) BIN FIXED(31)) EXTERNAL;                              1    9
        DCL LOCATN CHAR(8) EXTERNAL;                                       1   10
        DCL  DEMAND FILE RECORD KEYED ENV (REGIONAL(3)),                   1   11
         SPACED FILE RECORD KEYED  ENV(REGIONAL(3));                       1   11
/*                                                                  */     1   12
/*                                                                  */     1   12
/*  ******************************************************************/    1   12
/*  ********************** ON CONDITION UNITS **********************/    1   12
/*                                                                  */     1   12
        ON KEY(DEMAND)                                                     1   12
           BEGIN;                                                          2   13
           IE=6;                                                           2   14
           CALL ERRORM(P,IE,TOTAL_CARDS1,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,   2   15
           STPCSC,STPCSU);                                                 2   15
           GO TO ONE;                                                      2   16
           END;                                                            2   17
        ON KEY(SPACED)                                                     1   18
           BEGIN;                                                          2   19
           IE=7;                                                           2   20
           CALL ERRORM(P,IE,TOTAL_CARDS1,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,   2   21
           STPCSC,STPCSU);                                                 2   21
           GO TO ONE;                                                      2   22
           END;                                                            2   23
/*                                                                  */     1   24
        ON SUBSCRIPTRANGE                                                  1   24
           BEGIN;                                                          2   25
           LOCATN='    MAIN';                                              2   26
           IE=1;                                                           2   27
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

            CALL ERROR(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,STPCS   2   28
            C,STPCSU);                                                      2   28
               IF ISTOP=0 THEN                                              3   29
               GO TO END;                                                   3   30
            PUT SKIP EDIT('HOWEVER THIS BEING A DEBUGGING RUN THE EXECUTI   2   31
            ON WILL BE FORCED')(A);                                         2   31
            END;                                                            2   32
         ON CONVERSION                                                      1   33
            BEGIN;                                                          2   34
            IE=8;                                                           2   35
               IF FLAGC THEN                                                3   36
                  DO;                                                       4   37
                  CALL ERROR(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS    4   38
                  ,STPCSC,STPCSU);                                          4   38
                  GO TO END;                                                4   39
                  END;                                                      4   40
            CALL ERROR(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,STPCS    2   41
            C,STPCSU);                                                      2   41
            INDICATOR1=INDICATOR1+12;                                       2   42
            GO TO TWO;                                                      2   43
            END;                                                            2   44
/*                                                                  */     1   45
/*                                                                  */     1   45
         ON ENDFILE(INPUT1)                                                 1   45
            BEGIN;                                                          2   46
               IF FLAG5 THEN                                                3   47
               CALL UNVPRNT(ISTOP,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF   3   48
               ,NLOAD,PCS,P,LEUNIV,STPCSU,DEMASF,DALOAD,UPNASF,TOTALU,TA    3   48
               BLE3,ROOM,DESCRIP,LOAD_FACTOR);                              3   48
               IF FLAG6 THEN                                                3   49
               CALL PCSPRNT(PCSSWITCH);                                     3   50
            PUT SKIP(2)EDIT(' NORMAL END OF THE PROGRAM')(A);               2   51
            GO TO END;                                                      2   52
            END;                                                            2   53
/*                                                                  */     1   54
         ON ENDPAGE(DEPSUM)                                                 1   54
            BEGIN;                                                          2   55
            PUT PAGE FILE(DEPSUM)EDIT('KANSAS STATE UNIVERSITY :  DEPARTM   2   56
            ENT  SPACE SUMMARY')(COL(32),A);                                2   56
            PUT SKIP(2)FILE(DEPSUM)EDIT('REGIS DEPT','DESCRIPTION','ROOM-   2   57
              PCS ACTIVITY UNIT   SPACE-  PROJECTED   EXISTING')(COL(19)    2   57
            ,A,COL(36),A,COL(53),A)('CODE','TYPE','LOAD','FACTOR','NASF',   2   57
            'NASF')(SKIP,COL(26),A,COL(53),A,COL(69),A,COL(81),A,COL(94),   2   57
            A,COL(105),A);                                                  2   57
            PUT SKIP(2)FILE(DEPSUM);                                        2   58
            END;                                                            2   59
/*  ****************************************************************/     1   60
/*                                                                  */     1   60
/*  FIND OUT WHAT ALL SUMMARIES ARE REQUIRED                        */     1   60
         GET FILE(CONTROL)EDIT(COL_SUMMARY,UNIV_SUMMARY,PCS_SUMMARY,ISTOP   1   60
         )(A(4),A(4),A(4),F(4));                                            1   60
         FLAG4=COL_SUMMARY¬='    ';                                         1   61
         FLAG5=UNIV_SUMMARY¬='    ';                                        1   62
         FLAG6=PCS_SUMMARY¬='    ';                                        1   63
/*                   PRINT TITLE PAGE                               */     1   64
         CALL TITLES(FLAG4,FLAG5,FLAG6);                                    1   64
/*                                                                  */     1   65
         NR=13;                                                             1   65
```

```
        NC=10;                                                          1   66
        NU=30;                                                          1   67
        NL=05;                                                          1   68
/*  INITIALIZE ALL STORAGE AREAS                                */      1   69
        STPCSC=0;                                                       1   69
        CENASF=0;                                                       1   70
        CALOAD=0;                                                       1   71
        CPNASF=0;                                                       1   72
        TOTAL=0;                                                        1   73
        STPCSU=0;                                                       1   74
        UENASF=0;                                                       1   75
        UALOAD=0;                                                       1   76
        UPNASF=0;                                                       1   77
        TOTALU=0;                                                       1   78
        LENCOL,LENUNIV=0;                                               1   79
        PPNASF,PENASF=0;                                                1   80
        GET SKIP FILE(CONTROL)EDIT(TABLE3)(F(4));                       1   81
/*  OPEN ALL FILES                                              */      1   82
        OPEN FILE(DEMAND)DIRECT INPUT,FILE(SPACED)DIRECT INPUT,FILE(INP 1   82
        UT1)RECORD SEQUENTIAL INPUT,FILE(DEPSUM)PRINT PAGESIZE(59),FILE( 1  82
        COLSUMA)PRINT PAGESIZE(59),FILE(COLSUMB)PRINT PAGESIZE(59)LINESI 1  82
        ZE(132),FILE(UNIVSUM)PRINT PAGESIZE(59),FILE(PCSSUM)PRINT PAGESI 1  82
        ZE(59);                                                         1   82
/*  ALL ERROR MESSAGES PRINTED IN SYSPRINT                      */      1   83
        PUT FILE(SYSPRINT)EDIT('ERROR MESSAGES FOR THIS RUN')(COL(54),A) 1  83
        ;                                                               1   83
        PUT SKIP(4)FILE(SYSPRINT);                                      1   84
        SIGNAL ENDPAGE(DEPSUM);                                         1   85
/*  READ IN ALL DEFINITIONS, ETC.                               */      1   86
        GET FILE(SYSIN)EDIT((DESCRIP(IR)DO IR=1 TO NR))(A(20));         1   86
        GET SKIP FILE(SYSIN)EDIT((ROOM(IR)DO IR=1 TO NR))(X(1),A(3));   1   87
        UNIT(0)='   ';                                                 1   88
        GET SKIP FILE(SYSIN)EDIT((UNIT(LF)DO LF=1 TO 30))(X(1),A(3));   1   89
        ROOMSUB(0)=' ##';                                               1   90
        GET SKIP FILE(SYSIN)EDIT((ROOMSUB(LF)DO LF=1 TO 30))(X(1),A(3)); 1  91
        GET SKIP FILE(SYSIN)EDIT((PCSDEF(JJ)DO JJ=1 TO 8))(A(20));      1   92
        GET SKIP FILE(SYSIN)EDIT((PCSVAL(JJ)DO JJ=1 TO 40))(F(4));      1   93
        CARDCOUNT1=0;                                                   1   94
        FLAG1='0'B;                                                     1   95
        ICOL,IUNIV=1;                                                   1   96
ONE:    READ FILE(INPUT1)SET(P);                                        1   97
        INDICATOR1=P;                                                   1   98
        GET STRING(CARD1)EDIT(ADM_CODE)(A(4));                          1   99
           IF ADM_CODE='NEW ' THEN                                      2  100
              DO;                                                       3  101
/*  READ IN COLLEGE INFO LIKE  HEGIS, ADMN-CODE, COLLEGE NAME  */      3  102
              GET STRING(SUBSTR(CARD1,5,28))EDIT(ADMC,HEGISC,COLLEGE)(A( 3  102
              4),A(4),A(20));                                           3  102
              GO TO ONE;                                                3  103
              END;                                                      3  104
/*  IF THE DATA FOR CURRENT COLLEGE IS OVER THEN CALL COLPRNT   */      2  105
           IF ADM_CODE='OVER' THEN                                      2  105
              DO;                                                       3  106
                 IF FLAG4 THEN                                          4  107
                 CALL COLPRNT(ISTOP,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXN 4  108
                 ASF,NLOAD,PCS,P,LENCOL,STPCSC,CENASF,CALOAD,CPNASF,TOTA 4  108
                 L,TABLE3,ROOM,DESCRIP,LOAD_FACTOR);                    4  108
              GO TO ONE;                                                3  109
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

            END;                                                          3  110
/*  PROCESS THIS CARD                                                  */  1  111
        GET STRING(SUBSTR(CARD1,5,2))EDIT(CARD#1,TOTAL_CARDS1)(F(1),F(1)  1  111
        );                                                                1  111
        CARDCOUNT1=CARDCOUNT1+1;                                          1  112
/*  CHECK FOR CARDS OUT OF SEQUENCE                                    */  2  113
          IF CARDCOUNT1¬=CARD#1 THEN                                      2  113
             DO;                                                          3  114
             IE=3;                                                        3  115
             LOCATN='    MAIN';                                           3  116
             CALL ERRORM(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,ST   3  117
             PCSC,STPCSU);                                                3  117
             GO TO END;                                                   3  118
             END;                                                         3  119
          IF CARD#1=1 THEN                                                2  120
             DO;                                                          3  121
                IF ADM_CODE=STADM_CODE THEN                               4  122
                   DO;                                                    5  123
                   IE=3;                                                  5  124
                   LOCATN='    MAIN';                                     5  125
                   CALL ERRORM(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,   5  126
                   PCS,STPCSC,STPCSU);                                    5  126
                   GO TO END;                                             5  127
                   END;                                                   5  128
             STADM_CODE=ADM_CODE;                                         3  129
             FLAGC='1'B;                                                  3  130
/*  LOCATE RECORDS IN THE ACTIVITY LOAD FILE AND EXISTING SPACE FILE   */  3  131
             GET STRING(ADM_CODE)EDIT(INTER)(F(4));                       3  131
                IF INTER<1499 THEN                                        4  132
                SUBSTR(REGION,8,1)='0';                                   4  133
                ELSE                                                      4  134
                   IF INTER<1999 THEN                                     5  134
                   SUBSTR(REGION,8,1)='1';                                5  135
                   ELSE                                                   5  136
                      IF INTER<3999 THEN                                  6  136
                      SUBSTR(REGION,8,1)='2';                             6  137
                      ELSE                                                6  138
                      SUBSTR(REGION,8,1)='3';                             6  138
             READ FILE(DEMAND)INTO(DEMAND_DATA)KEY(ADM_CODE||REGION);     3  139
             READ FILE(SPACED)INTO(EXSPDATA)KEY(ADM_CODE||REGION);        3  140
             PUT SKIP(2)FILE(SPRSUM)EDIT(DEMAND_DATA.REGIS,ADM_CODE,DEP   3  141
             T_NAME)(COL(19),A,COL(25),A,COL(31),A);                      3  141
             STPCSCODE,STROOM_CODE=0;                                     3  142
             FLAG3='0'B;                                                  3  143
             END;                                                         3  144
/*  CHECK FOR INPUT CARDS BEING OUT OF SEQUENCE                        */  2  145
          IF CARD#1¬=1 THEN                                               2  145
             IF ADM_CODE¬=STADM_CODE THEN                                 3  146
                DO;                                                       4  147
                IE=3;                                                     4  148
                LOCATN='    MAIN';                                        4  149
                CALL ERRORM(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS   4  150
                ,STPCSC,STPCSU);                                          4  150
                GO TO END;                                                4  151
                END;                                                      4  152
TWO:        IF INDICATOR1=30 THEN                                         2  153
            GO TO ONE;                                                    2  154
        FLAGC='0'B;                                                       1  155
```

```
            IF FLAG1 THEN                                                    2  156
               DO;                                                           3  157
               FLAG1='0'B;                                                   3  158
               GET STRING(SUBSTR(CARD1,INDICATOR1,77))EDIT(LOAD_FACTOR,ACT   3  159
               IVITY_LOAD)(F(2),F(5,1));                                     3  159
               FLAG1A='1'B;                                                  3  160
               INDICATOR1=INDICATOR1+12;                                     3  161
               GO TO PROCESS;                                                3  162
               END;                                                          3  163
         DATA1=SUBSTR(CARD1,INDICATOR1,11);                                  1  164
         GET STRING(DATA1)EDIT(ROOM_CODE,LOAD_FACTOR,SPACE_FACTOR,PCS)(F(    1  165
         2),F(2),F(5,1),F(2));                                               1  165
         INDICATOR1=INDICATOR1+12;                                           1  166
/*   CHECK IF ALL DATA FOR THIS DEPT HAS BEEN PROCESSED */                   2  167
/*  THIS IS INDICATED ROOM CODE VALUE BEING ZERO                     */     2  167
            IF ROOM_CODE=0 THEN                                              2  167
               DO;                                                           3  168
               CARDCOUNT1=0;                                                 3  169
               SWITCH1='1'B;                                                 3  170
               CALL TOTLPRNT;                                                3  171
               PUT SKIP(2)FILE(DEPSUM)EDIT('GRAND-TOTAL',GRANDTOTAL1,GRAN    3  172
               DTOTAL2)(COL(40),A,COL(80),P'(10)Z',COL(99),P'(10)Z');        3  172
               GRANDTOTAL1,GRANDTOTAL2=0;                                    3  173
                  IF CARD#1¬=TOTAL_CARDS1 THEN                               4  174
                     DO;                                                     5  175
                     IE=3;                                                   5  176
                     LOCATN='    MAIN';                                      5  177
                     CALL ERROR#(P,IE,KFACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,    5  178
                     PCS,STPCSC,STPCSU);                                     5  178
                     GO TO END;                                              5  179
                     END;                                                    5  180
               GO TO ONE;                                                    3  181
               END;                                                          3  182
/*    SKIP  DUMMY DATA   */                                                 2  183
            IF ROOM_CODE=99 THEN                                             2  183
            GO TO TWO;                                                       2  184
/*  LOAD FACTOR =99 IMPLIES THAT THE DEMAND DATA IS ON CARDS  */             2  185
            IF LOAD_FACTOR=99 THEN                                           2  185
               DO;                                                           3  186
               FLAG1='1'B;                                                   3  187
               GO TO TWO;                                                    3  188
               END;                                                          3  189
/*                                                                  */       1  190
/*                                                                  */       1  190
/*  PROCESS THE CARD FOR ANY ERRORS LIKE IMPROPER PCS, LOAD FACTOR-  */      1  190
/*  ETC.                                                            */      1  190
PROCESS:                                                                     1  190
            DO IPC=1 TO 40;                                                  2  190
               IF PCSVAL(IPC)=PCS THEN                                       3  191
               GO TO CARRYON;                                                3  192
            END;                                                             2  193
/*    THIS IS AN ERROR CONDITION  */                                         1  194
         IE=4;                                                               1  194
         LOCATN='    MAIN';                                                  1  195
         CALL ERROR#(P,IE,LOAD_FACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,STPC    1  196
         SC,STPCSU);                                                         1  196
         GO TO TWO;                                                          1  197
/*  LOAD FACTOR = 0 IMPLIES ACTIVITY LOAD IS ZERO */                        1  198
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

CARRYON:KFACTOR=LOAD_FACTOR;                                          1  198
        IF LOAD_FACTOR=10 THEN                                        2  199
        KFACTOR=15;                                                   2  200
        IF LOAD_FACTOR=15 THEN                                        2  201
        KFACTOR=16;                                                   2  202
        IF LOAD_FACTOR=20 THEN                                        2  203
        KFACTOR=17;                                                   2  204
        DO NLOAD=1 TO NL;                                             2  205
           IF TABLE3(ROOM_CODE,NLOAD)=KFACTOR THEN                    3  206
           GO TO COMP;                                                3  207
        END;                                                          2  208
/*   THIS IS AN ERROR CONDITION   */                                  1  209
     IE=5;                                                            1  209
     LOCATN='    MAIN';                                               1  210
     CALL ERROR1(P,IE,LOAD_FACTOR,NLOAD,ICOL,IUNIV,ROOM_CODE,PCS,STPC 1  211
     SC,STPCSU);                                                      1  211
     GO TO TWO;                                                       1  212
COMP:   IF LOAD_FACTOR=0 THEN                                         2  213
        ACTIVITY_LOAD=0.;                                             2  214
        ELSE                                                          2  215
           IF¬FLAG1A THEN                                             3  215
           ACTIVITY_LOAD=DATA_TYPE(LOAD_FACTOR);                      3  216
     FLAG1A='0'B;                                                     1  217
        IF ROOM_CODE=1 THEN                                           2  218
        ACTIVITY_LOAD=ACTIVITY_LOAD*10.;                              2  219
/*                                                                 */ 1  220
     SWITCH1=ROOM_CODE¬=STROOM_CODE;                                  1  220
     SWITCH2=PCS¬=STPCSCODE;                                          1  221
        IF SWITCH1|SWITCH2 THEN                                       2  222
           DO;                                                        3  223
              IF FLAG3 THEN                                           4  224
              CALL TOTLPRNT;                                          4  225
           FLAG3='1'B;                                                3  226
           EXNASF=0;                                                  3  227
              IF LOAD_FACTOR=0 THEN                                   4  228
                 DO;                                                  5  229
                    IF SWITCH1 THEN                                   6  230
                    FLAG2='1'B;                                       6  231
                    ELSE                                              6  232
                    FLAG2A='1'B;                                      6  232
                 PROJNASF=0;                                          5  233
                 END;                                                 5  234
              DO KSEARCH=1 TO 14;                                     4  235
                 IF EXROOM_CODE(KSEARCH)=ROOM_CODE THEN               5  236
                    IF EXPCS(KSEARCH)=PCS THEN                        6  237
                       DO;                                            7  238
                       EXNASF=EXISTING_SPACE(KSEARCH);                7  239
                       GO TO EXIT1;                                   7  240
                       END;                                           7  241
              END;                                                    4  242
           END;                                                       3  243
EXIT1:  IF LOAD_FACTOR¬=0 THEN                                        2  244
        PROJNASF=ACTIVITY_LOAD*SPACE_FACTOR;                          2  245
        IF¬SWITCH1 THEN                                               2  246
           IF¬SWITCH2 THEN                                            3  247
              IF LOAD_FACTOR=0 THEN                                   4  248
              DO;                                                     5  249
/*             THIS IS AN ERROR CONDITION  */                         5  250
```

```
(SU'S PL/I NEATENER AND PRECOMPILER                    PAGE

                    IE=5;                                                    5  250
                    LOCATN='   MAIN';                                        5  251
                    CALL ERROR1(P,IE,LOAD_FACTOR,NLOAD,ICOL,IUNIV,ROOM_C     5  252
                    ODE,PCS,STPCSC,STPCSU);                                  5  252
                    GO TO TWO;                                               5  253
                    END;                                                     5  254
              IF LOAD_FACTOR=0 THEN                                          2  255
              GO TO CHECK1;                                                  2  256
              IF FLAG2 THEN                                                  2  257
                 DO;                                                         3  258
                 PUT SKIP(2)FILE(DEPSUM)EDIT(DESCRIP(ROOM_CODE),ROOM(ROOM_C  3  259
                 ODE)||' '||ROOMSUB(LOAD_FACTOR),PCS,ACTIVITY_LOAD,UNIT(LOA  3  259
                 D_FACTOR),SPACE_FACTOR,PROJNASF)(COL(31),A,COL(53),A,COL(6  3  259
                 1),F(3),COL(65),F(5,2),COL(75),A,COL(80),F(7,2),COL(88),P'  3  259
                 (8)Z(2)9');                                                 3  259
                 FLAG2='0'B;                                                 3  260
                 TOTAL1=TOTAL1+PROJNASF;                                     3  261
                 GO TO CHECK1;                                               3  262
                 END;                                                        3  263
           PUT SKIP FILE(DEPSUM)EDIT(ROOMSUB(LOAD_FACTOR),PCS,ACTIVITY_LOAD  1  264
           ,UNIT(LOAD_FACTOR),SPACE_FACTOR,PROJNASF)(COL(57),A,COL(61),F(3)  1  264
           ,COL(65),F(5,2),COL(75),A,COL(80),F(7,2),COL(88),P'(8)Z(2)9');    1  264
           TOTAL1=TOTAL1+PROJNASF;                                           1  265
CHECK1:    ICOLS=ISTOP;                                                      1  266
           IUNIS=ISTOP;                                                      1  267
              IF FLAG4 THEN                                                  2  268
              CALL COLCHK(ICOLS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,NLO  2  269
              AD,PCS,P,LENCOL,STPCSC,GENASF,CALOAD,CPNASF,TOTAL,TABLE3,ROOM  2  269
              ,DESCRIP,LOAD_FACTOR);                                         2  269
              IF FLAG5 THEN                                                  2  270
              CALL UNIVCHK(IUNIS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,NL  2  271
              OAD,PCS,P,LENUNIV,STPCSU,UENASF,UALOAD,UPNASF,TOTALU,TABLE3,R  2  271
              OOM,DESCRIP,LOAD_FACTOR);                                      2  271
/*                                                                  */       2  272
/*                                                                  */       2  272
/*                                                                   */      2  272
/*.   UPDATE  PCS TOTALS  IF REQUIRED    */                                 2  272
              IF FLAG6 THEN                                                  2  272
                 DO;                                                         3  273
                    IF LOAD_FACTOR=0 THEN                                    4  274
                    PPNASF(IPC)=PPNASF(IPC)+EXNASF;                          4  275
                    ELSE                                                     4  276
                    PPNASF(IPC)=PPNASF(IPC)+PROJNASF;                        4  276
                 PCSSWITCH(PCS/10)='1'B;                                     3  277
                    IF SWITCH1|SWITCH2 THEN                                  4  278
                    PENASF(IPC)=PENASF(IPC)+EXNASF;                          4  279
                 END;                                                        3  280
/*   RESET RELEVANT QUANTITIES */                                           1  281
           STPCSCODE=PCS;                                                    1  281
           STROOM_CODE=ROOM_CODE;                                            1  282
           GO TO TWO;                                                        1  283
/*                                                                  */       2  284
/*                                                                  */       2  284
/*                                                          */               2  284
TOTLPRNT:  PROCEDURE;                                                        2  284
/*  ************************************************************************/  2  285
/*                                                                    */     2  285
/*  THIS IS AN INTERNAL PROCEDURE WHICH PRINTS THE SUBTOTALS FOR EACH*/      2  285
```

```
KSU'S PL/I CATENER AND PRECOMPILER                    PAGE

/*  ROOM TYPE AND A PARTICULAR POS, TOTAL FOR A PARTICULAR ROOM TYPE */   2  285
/*  AND ALL POS ASSOCIATED WITH IT                                   */   2  285
/*  UPDATES TOTAL, TOTALU, CENASF & UENASF                           */   2  285
/*  UPDATES THE FINAL TOTALS (PROJECTED AND EXISTING) FOR A DEPT     */   2  285
/*                                                                   */   2  285
/*  *****************************************************************/   2  285
      DCL (SUBTOTAL1,SUBTOTAL2) STATIC BIN FIXED(31) INIT (0);           2  285
          ON CONVERSION SYSTEM;                                          2  286
             IF FLAG2A THEN                                              3  287
                DO;                                                      4  288
                FLAG2A='0'B;                                             4  289
                TOTAL1=EXNASF;                                           4  290
                PUT SKIP FILE(DEPSUM)EDIT('##',STPOSCODE,TOTAL1,EXNASF)  4  291
                (COL(58),A,COL(61),F(3),COL(88),P'(8)Z(2)9',COL(99),P'(  4  291
                8)Z(2)9');                                               4  291
                GO TO OUT;                                               4  292
                END;                                                     4  293
             IF FLAG2 THEN                                               3  294
                DO;                                                      4  295
                FLAG2='0'B;                                              4  296
                TOTAL1=EXNASF;                                           4  297
                PUT SKIP(2)FILE(DEPSUM)EDIT(DESCRIP(STROOM_CODE),ROOM(S  4  298
                TROOM_CODE),'##',STPOSCODE,TOTAL1,EXNASF)(COL(31),A,COL  4  298
                (53),A,COL(58),A,COL(61),F(3),COL(88),P'(8)Z(2)9',COL(9  4  298
                9),P'(8)Z(2)9');                                         4  298
                GO TO OUT;                                               4  299
                END;                                                     4  300
             IF TOTAL1=PROJNASF THEN                                     3  301
             PUT SKIP(0)FILE(DEPSUM)EDIT(EXNASF)(COL(99),P'(8)Z(2)9');   3  302
             ELSE                                                        3  303
             PUT SKIP FILE(DEPSUM)EDIT('SUB-TOT',STPOSCODE,TOTAL1,EXNAS  3  303
             F)(COL(53),A,COL(61),F(3),COL(88),P'(8)Z',COL(099),P'(8)Z   3  303
             (2)9');                                                     3  303
OUT:      SUBTOTAL1=TOTAL1+SUBTOTAL1;                                    2  304
          SUBTOTAL2=SUBTOTAL2+EXNASF;                                    2  305
             IF FLAG4 THEN                                               3  306
                DO;                                                      4  307
                TOTAL(STROOM_CODE,ICOLS)=TOTAL1+TOTAL(STROOM_CODE,ICOLS  4  308
                );                                                       4  308
                CENASF(STROOM_CODE,ICOLS)=CENASF(STROOM_CODE,ICOLS)+EXN  4  309
                ASF;                                                     4  309
                END;                                                     4  310
             IF FLAG5 THEN                                               3  311
                DO;                                                      4  312
                TOTALU(STROOM_CODE,IUNIS)=TOTAL1+TOTALU(STROOM_CODE,IUN  4  313
                IS);                                                     4  313
                UENASF(STROOM_CODE,IUNIS)=UENASF(STROOM_CODE,IUNIS)+EXN  4  314
                ASF;                                                     4  314
                END;                                                     4  315
             IF STROOM_CODE¬=1 THEN                                      3  316
             GRANDTOTAL1=GRANDTOTAL1+TOTAL1;                             3  317
          GRANDTOTAL2=GRANDTOTAL2+EXNASF;                                2  318
             IF¬SWITCH1 THEN                                             3  319
                DO;                                                      4  320
                TOTAL1=0;                                                4  321
                RETURN;                                                  4  322
                END;                                                     4  323
             IF SUBTOTAL1=TOTAL1 THEN                                    3  324
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

                IF SUBTOTAL2=EXNASE THEN                                    4  325
                   DO;                                                      5  326
                   SUBTOTAL1,SUBTOTAL2=0;                                   5  327
                   TOTAL1=0;                                                5  328
                   FLAG2='1'B;                                              5  329
                   RETURN;                                                  5  330
                   END;                                                     5  331
          PUT SKIP FILE(DEPSUM)EDIT('TOT',SUBTOTAL1,SUBTOTAL2)(COL(57),     2  332
          A,COL(89),P'(10)Z',COL(99),P'(8)Z(2)9');                          2  332
          SUBTOTAL1,SUBTOTAL2,TOTAL1=0;                                     2  333
          FLAG2='1'B;                                                       2  334
          END TOTLPRNT;                                                     2  335
END:    END SPPROJ;                                                         1  336
```

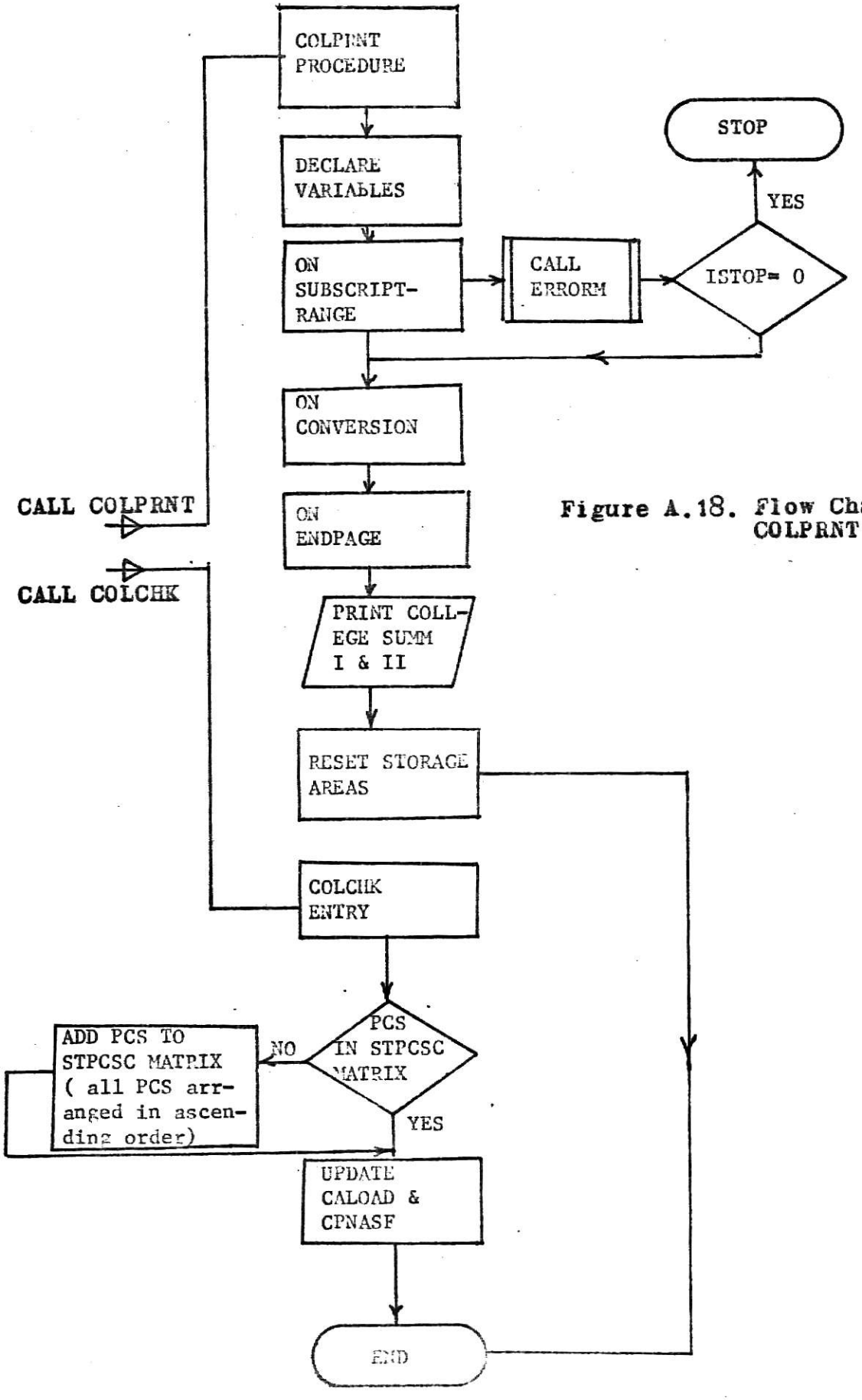


Figure A.18. Flow Chart of-COLPRNT

Figure A.19. COLPRNT: Source Listings

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

(SUBSCRIPTRANGE):COLPRNT:                                                    1     1
        PROCEDURE(ICOLS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,NLOAD,PC     1     1
        S,J,LENCOL,STPCSC,CENASF,CALOAD,CPNASF,TOTAL,TABLE3,ROOM,DESCRIP     1     1
        ,LOAD_FACTOR);                                                       1     1
/*                                                                        */ 1     2
/*************************************************************************/ 1     2
/*  THIS PROCEDURE CONSISTS OF TWO SECTIONS : COLPRNT & COLCHK            */ 1     2
/*  COLPRNT IS THE MAIN ENTRY POINT, AND THIS SECTION OF THE              */ 1     2
/*  PROCEDURE ACCOMPLISHES THE FOLLOWING :-                               */ 1     2
/*        1. PRINTS COLLEGE SUMMARIES I & II                              */ 1     2
/*        2. RESETS STORAGE AREAS                                         */ 1     2
/*  COLCHK IS THE SECONDARY ENTRY POINT AND THIS SECTION OF THE           */ 1     2
/*  PROCEDURE ACCOMPLISHES THE FOLLOWING :-                               */ 1     2
/*        1. STORES THE PCS VALUES IN THE MATRIX STPCSC                   */ 1     2
/*        2. UPDATES ALL TOTALS                                           */ 1     2
/*  ..................... DEFINITIONS OF IMPORTANT VARIABLES ......      */ 1     2
/*  SUBTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR A PARTICULAR           */ 1     2
/*  AND ALL ASSOCIATED PCS                                                */ 1     2
/*  SUBTOTAL2 : ACCUMULATES THE EXISTING NASF FOR A PARTICULAR ROOM-      */ 1     2
/*  TYPE AND ALL ASSOCIATED PCS                                           */ 1     2
/*  GRANDTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR THE ENTIRE           */ 1     2
/*  COLLEGE                                                               */ 1     2
/*  GRANDTOTAL2 : ACCUMULATES THE EXISTING NASF FOR THE ENTIRE            */ 1     2
/*  COLLEGE                                                               */ 1     2
/*  FLAG1 : SWITCH USED TO CONTROL OUTPUT PRINTOUT                        */ 1     2
/*  ALL OTHER VARIABLES HAVE BEEN DEFINED IN THE MAIN PROCEDURE           */ 1     2
/*                                                                        */ 1     2
/*************************************************************************/ 1     2
                                                                      DC L   1     2
 ICOLS, ACTIVITY_LOAD FIXED DEC(7,2), ROOM_CODE BIN FIXED,             PR O   1     2
JNASF BIN FIXED(31), EXNASF BIN FIXED(31),NLOAD,PCS BIN FIXED;               1     2
        DCL LENCOL(*), STPCSC(*,*) BIN FIXED, CENASF(*,*) BIN FIXED(31)      1     3
,       CALOAD(*,*,*) FIXED DEC(10,2), CPNASF(*,*,*) BIN FIXED(31),          1     3
        TOTAL(*,*) BIN FIXED(31), ROOM(*) CHAR(*), DESCRIP(*) CHAR(*),       1     3
        TABLE3(*,*) BIN FIXED;                                               1     3
        DCL ( ROOMSUB (0:30)    CHAR(3), UNIT (0:30) CHAR(3)) EXTERNAL;      1     4
        DCL (NR,NC,NU,NL) EXTERNAL;                                          1     5
        DCL (ADMC CHAR(4), REGISC CHAR(4), COLLEGE CHAR(20)) EXTERNAL;       1     6
        DCL (TOTALP(13),TOTALE(13)) FIXED BIN (31) INIT ((13)0),             1     7
        DEF BIN FIXED,          FLAG1 BIT(1) INIT ('1'B),                    1     7
            (SUBTOTAL1,SUBTOTAL2) BIN FIXED(31) INIT (0),                    1     7
            (GRANDTOTAL1,GRANDTOTAL2) BIN FIXED (31) INIT (0), CHECKTOT      1     7
AL      BIN FIXED(31) INIT (0), CHEK BIT(1) STATIC INIT ('1'B), Q POINT      1     7
ER;                                                                          1     7
        DCL LOCATN CHAR(8) EXTERNAL;                                         1     8
        DCL DUMMY (1,1) BIN FIXED INIT(0);                                   1     9
        ON CONVERSION SYSTEM;                                                1    10
        ON SUBSCRIPTRANGE                                                    1    11
           BEGIN;                                                            2    12
           IE=1;                                                             2    13
           LOCATN=' COLPRNT';                                                2    14
           KFACTOR=88;                                                       2    15
           CALL EPROGM(Q,IE,KFACTOR,NLOAD,ICOL,IUNIV,IP,PCS,STPCSC,DUMMY     2    16
           );                                                                2    16
              IF ICOLS=0 THEN                                                3    17
              STOP;                                                          3    18
           PUT SKIP EDIT('HOWEVER THIS BEING A DEBUGGING RUN THE EXECUTI     2    19
           ON WILL BE FORCED')(A);                                           2    19
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

          END;                                                          2    20
       ON ENDPAGE(COLSUMA)                                              1    21
          BEGIN;                                                        2    22
          PUT PAGE FILE(COLSUMA)EDIT('KANSAS STATE UNIVERSITY :  COLLEG 2    23
          E SPACE SUMMARY  I')(COL(35),A);                              2    23
          PUT SKIP(3)FILE(COLSUMA)EDIT(COLLEGE,'REGIS:  ',REGISC)(COL(4 2    24
          0),A,COL(65),A,X);                                            2    24
          PUT SKIP(3)FILE(COLSUMA)EDIT('DESCRIPTION','ROOM-   PCS ACTIV 2    25
          ITY UNIT  PROJECTED   EXISTING')(COL(31),A,COL(53),A)('TYPE',  2    25
          'LOAD','NASF','NASF')(SKIP,COL(55),A,COL(69),A,COL(85),A,COL( 2    25
          96),A);                                                       2    25
          PUT SKIP(2)FILE(COLSUMA);                                     2    26
          END;                                                          2    27
       ON ENDPAGE(COLSUMB)                                              1    28
          BEGIN;                                                        2    29
          PUT PAGE FILE(COLSUMB)EDIT('KANSAS STATE UNIVERSITY :  COLLEG 2    30
          E SPACE SUMMARY  II')(COL(32),A);                             2    30
          PUT SKIP(2)FILE(COLSUMB)EDIT('NOTE :  THE DEF / SUR NASF IS I 2    31
          N HUNDREDS  OF SQUARE FEET')(COL(16),A)('THE % AGE SURPLUS (O 2    31
          R DEF) HAS BEEN  EXPRESSED AS A PERCENTAGE OF THE EXISTING NA 2    31
          SF')(SKIP,COL(24),A);                                         2    31
          PUT SKIP(2)FILE(COLSUMB)EDIT('REGIS ADMN-      COLLEGE','200 2    32
          ','250','300','400','500','600','700','800-900')(COL(6),A,X(1 2    32
          1),A,X(8),A,X(8),A,X(8),A,X(8),A,X(8),A,X(8),A,X(4),A)('CODE' 2    32
          ,REPEAT(' SUR-   % ',7))(SKIP,COL(12),A,COL(37),A);           2    32
          PUT SKIP(2)FILE(COLSUMB);                                     2    33
          END;                                                          2    34
/*                                                                 */   1    35
       SIGNAL ENDPAGE(COLSUMA);                                         1    35
          IF CHEK THEN                                                  2    36
          SIGNAL ENDPAGE(COLSUMB);                                      2    37
       CHEK='0'B;                                                       1    38
RMC:      DO IR=1 TO NR;                                                2    39
PCC:         DO IP=1 TO NC;                                             3    40
                IF STPCSC(IR,IP)=0 THEN                                 4    41
                GO TO EXTC;                                             4    42
                IF TOTAL(IR,IP)=0 THEN                                  4    43
                   IF CENASF(IR,IP)=0 THEN                              5    44
                   GO TO CONTP;                                         5    45
             KOUNT=0;                                                   3    46
LFC:            DO IL=1 TO NL;                                          4    47
                JJ=TABLE3(IR,IL);                                       4    48
                   IF CPNASF(IR,IP,IL)=0 THEN                           5    49
                   GO TO CONTL;                                         5    50
                KOUNT=KOUNT+1;                                          4    51
                   IF FLAG1 THEN                                        5    52
                      DO;                                               6    53
                      FLAG1='0'B;                                       6    54
                      PUT SKIP(2)FILE(COLSUMA)EDIT(DESCRIP(IR),ROOM(IR) 6    55
                      ||' '||ROOMSUB(JJ),STPCSC(IR,IP),CALOAD(IR,IP,IL) 6    55
                      ,UNIT(JJ),CPNASF(IR,IP,IL))(COL(31),A,COL(53),A,C 6    55
                      OL(61),F(3),COL(65),F(3,1),COL(75),A,COL(79),P'(1 6    55
                      0)Z');                                            6    55
                      GO TO CONTL;                                      6    56
                      END;                                              6    57
                PUT SKIP FILE(COLSUMA)EDIT(ROOMSUB(JJ),STPCSC(IR,IP),CA 4    58
                LOAD(IR,IP,IL),UNIT(JJ),CPNASF(IR,IP,IL))(COL(57),A,COL 4    58
                (61),F(3),COL(65),F(3,1),COL(75),A,COL(79),P'(10)Z');   4    58
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

CONTL:            END LFC;                                          4   59
                  IF FLAG1 THEN                                     4   60
                     DO;                                            5   61
                     FLAG1='0'B;                                    5   62
                     PUT SKIP(2)FILE(COLSUMA)EDIT(DESCRIP(IR),ROOM(IR),ST  5   63
                     PCSC(IR,IP),TOTAL(IR,IP),CENASF(IR,IP))(COL(31),A,CO  5   63
                     L(53),A,COL(61),F(3),COL(79),P'(10)Z',COL(90),P'(8)Z  5   63
                     (2)9');                                        5   63
                     END;                                           5   64
                  ELSE                                              4   65
                     DO;                                            5   65
                        IF KOUNT=1 THEN                             6   66
                        PUT SKIP(2)FILE(COLSUMA)EDIT(CENASF(IR,IP))(COL(9  6   67
                        0),P'(8)Z(2)9');                            6   67
                        ELSE                                        6   68
                        PUT SKIP FILE(COLSUMA)EDIT('SUB-TOT',STPCSC(IR,IP  6   68
                        ),TOTAL(IR,IP),CENASF(IR,IP))(COL(53),A,COL(61),F  6   68
                        (3),COL(79),P'(10)Z',COL(90),P'(10)Z');     6   68
                     END;                                           5   69
               SUBTOTAL1=SUBTOTAL1+TOTAL(IR,IP);                    3   70
               SUBTOTAL2=SUBTOTAL2+CENASF(IR,IP);                   3   71
               TOTALP(IR)=TOTALP(IR)+TOTAL(IR,IP);                  3   72
               TOTALE(IR)=TOTALE(IR)+CENASF(IR,IP);                 3   73
                  IF IR¬=1 THEN                                     4   74
                  GRANDTOTAL1=GRANDTOTAL1+TOTAL(IR,IP);             4   75
               GRANDTOTAL2=GRANDTOTAL2+CENASF(IR,IP);               3   76
               GO TO CONTP;                                         3   77
EXTC:             IF IP=1|IP=2 THEN                                4   78
                     DO;                                            5   79
                     SUBTOTAL1,SUBTOTAL2=0;                         5   80
                     FLAG1='1'B;                                    5   81
                     GO TO CONTR;                                   5   82
                     END;                                           5   83
               PUT SKIP FILE(COLSUMA)EDIT('TOT',SUBTOTAL1,SUBTOTAL2)(COL(  3   84
               57),A,COL(79),P'(10)Z',COL(90),P'(10)Z');            3   84
               FLAG1='1'B;                                          3   85
               SUBTOTAL1,SUBTOTAL2=0;                               3   86
               GO TO CONTR;                                         3   87
CONTP:         END PCC;                                            3   88
CONTR:      END RPC;                                               2   89
         PUT SKIP(2)FILE(COLSUMA)EDIT('GRAND-TOTAL',GRANDTOTAL1,GRANDTOTA  1   90
         L2)(COL(49),A,COL(79),P'(10)Z',COL(90),P'(10)Z');          1   90
SUMRY2:  PUT SKIP(2)FILE(COLSUMA)EDIT(REGISC,ADMC,COLLEGE)(COL(6),A,COL(1  1   91
         2),A,COL(17),A);                                           1   91
         TOTALP(5)=TOTALP(5)+TOTALP(6)+TOTALP(7);                   1   92
         TOTALE(5)=TOTALE(5)+TOTALE(6)+TOTALE(7);                   1   93
         TOTALP(6)=TOTALP(8);                                       1   94
         TOTALE(6)=TOTALE(8);                                       1   95
         TOTALP(7)=TOTALP(9);                                       1   96
         TOTALE(7)=TOTALE(9);                                       1   97
         TOTALP(8)=TOTALP(10);                                      1   98
         TOTALE(8)=TOTALE(10);                                      1   99
         TOTALP(9)=TOTALP(11)+TOTALP(12);                           1  100
         TOTALE(9)=TOTALE(11)+TOTALE(12);                           1  101
         I=37;                                                      1  102
PLOOP:      DO J=2 TO 9;                                            2  103
               IF TOTALE(J)=0 THEN                                  3  104
                  DO;                                               4  105
```

```
KSU'S PL/I SCATENER AND PRECOMPILER                    PAGE

                  TOTALP(J)=-TOTALP(J)/100;                                    4  106
                  PUT SKIP(3)FILE(COLSUMB)EDIT(TOTALP(J),'*')(COL(I),F(5)      4  107
                  ,X(4),A);                                                    4  107
                  END;                                                         4  108
               ELSE                                                            3  109
                  DO;                                                          4  109
                  DEF=(TOTALE(J)-TOTALP(J))*100/TOTALE(J);                     4  110
                  TOTALP(J)=(TOTALE(J)-TOTALP(J))/100;                         4  111
                  PUT SKIP(3)FILE(COLSUMB)EDIT(TOTALP(J),DEF)(COL(I),F(5)      4  112
                  ,X(1),F(4));                                                 4  112
                  END;                                                         4  113
            I=I+11;                                                            2  114
            END PLOOP;                                                         2  115
/*    RESET STORAGE AREAS  */                                                  1  116
      CALOAD(*,*,*)=0.;                                                        1  116
      CPNASF(*,*,*)=0;                                                         1  117
      CENASF(*,*),TOTAL(*,*)=0;                                                1  118
      STPCSC(*,*)=0;                                                           1  119
      GO TO OVER;                                                              1  120
/*                                                                  */         1  121
/*                                                                  */         1  121
/*                                                                  */         1  121
COLCHK: ENTRY(ICOLS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,ALOAD,PCS,Q,      1  121
      LENCOL,STPCSC,CENASF,CALOAD,CPNASF,TOTAL,TABLE3,ROOM,DESCRIP,LOA       1  121
      D_FACTOR);                                                               1  121
      IR=ROOM_CODE;                                                            1  122
         DO ICOL=1 TO NC;                                                      2  123
            IF STPCSC(ROOM_CODE,ICOL)=PCS THEN                                 3  124
            GO TO TEST;                                                        3  125
            IF STPCSC(ROOM_CODE,ICOL)=0 THEN                                   3  126
               DO;                                                             4  127
               LENCOL(ROOM_CODE)=ICOL;                                         4  128
               STPCSC(ROOM_CODE,ICOL)=PCS;                                     4  129
               GO TO TEST;                                                     4  130
               END;                                                            4  131
            IF STPCSC(ROOM_CODE,ICOL)>PCS THEN                                 3  132
               DO;                                                             4  133
               LENGTH=LENCOL(ROOM_CODE);                                       4  134
               NTIMES=LENCOL(ROOM_CODE)-ICOL+1;                                4  135
                  DO KCOL=1 TO NTIMES;                                         5  136
                  STPCSC(ROOM_CODE,LENGTH+1)=STPCSC(ROOM_CODE,LENGTH);         5  137
                  TOTAL(ROOM_CODE,LENGTH+1)=TOTAL(ROOM_CODE,LENGTH);           5  138
                  CENASF(ROOM_CODE,LENGTH+1)=CENASF(ROOM_CODE,LENGTH);         5  139
                  CALOAD(ROOM_CODE,LENGTH+1,*)=CALOAD(ROOM_CODE,LENGTH         5  140
                  ,*);                                                         5  140
                  CPNASF(ROOM_CODE,LENGTH+1,*)=CPNASF(ROOM_CODE,LENGTH         5  141
                  ,*);                                                         5  141
                  LENGTH=LENGTH-1;                                             5  142
                  END;                                                         5  143
               LENCOL(ROOM_CODE)=LENCOL(ROOM_CODE)+1;                          4  144
               STPCSC(ROOM_CODE,ICOL)=PCS;                                     4  145
               CENASF(ROOM_CODE,ICOL),TOTAL(ROOM_CODE,ICOL)=0;                 4  146
               CALOAD(ROOM_CODE,ICOL,*)=0.;                                    4  147
               CPNASF(ROOM_CODE,ICOL,*)=0;                                     4  148
               GO TO TEST;                                                     4  149
               END;                                                            4  150
         END;                                                                  2  151
TEST:    IF LOAD_FACTOR=0 THEN                                                 2  152
```

```
          PROJNASF=EXNASF;                                                    2  153
       IN=ROOM_CODE;                                                          1  154
       CALOAD(IN,ICOL,NLOAD)=CALOAD(IN,ICOL,NLOAD)+ACTIVITY_LOAD;             1  155
       CPNASF(IN,ICOL,NLOAD)=CPNASF(IN,ICOL,NLOAD)+PROJNASF;                  1  156
       ICOLS=ICOL;                                                            1  157
OVER:  END COLPRNT;                                                           1  158
```

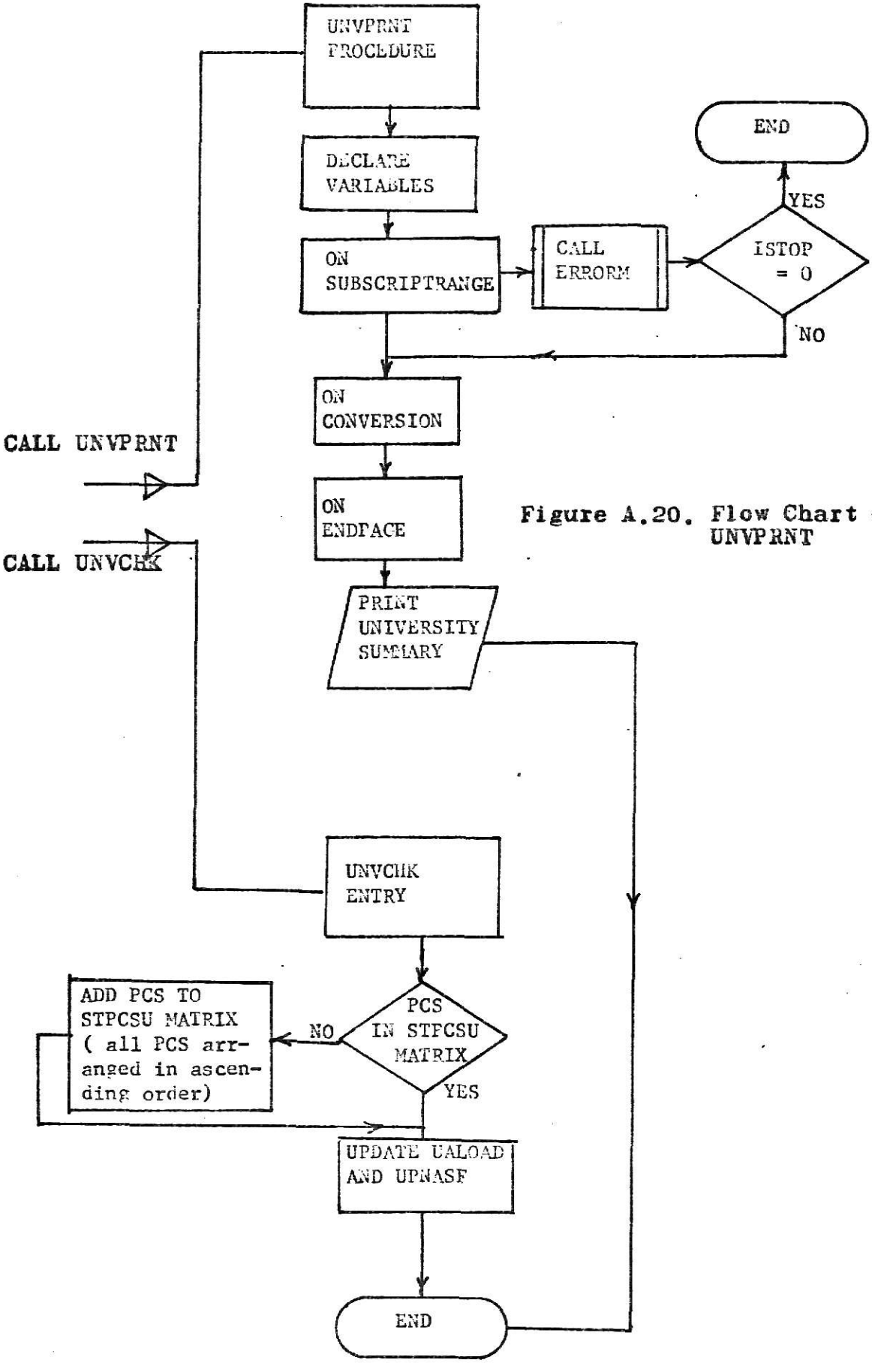


Figure A.20. Flow Chart of UNVPRNT

Figure A.21. UNVPRNT: Source Listings

```
KSU'S PL/I WATENER AND PRECOMPILER                    PAGE

(SUBSCRIPTRANGE):UNVPRNT:                                                    1    1
        PROCEDURE(IUNIS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,NLOAD,PC     1    1
        S,Q,LENUNIV,STPCSU,UENASF,UALOAD,UPNASF,TOTALU,TABLE3,ROOM,DESCR     1    1
        IP,LOAD_FACTOR);                                                     1    1
/*********************************************************************/     1    2
/*                                                                   */     1    2
/*  THIS PROCEDURE CONSISTS OF TWO SECTIONS : UNVCHK AND UNVPRNT     */     1    2
/*  UNVCHK IS AN ENTRY POINT IN THIS PROCEDURE ACCOMPLISHING THE     */     1    2
/*  FOLLOWING :-                                                     */     1    2
/*        1. STORES THE PCS VALUES IN THE MATRIX STPCSU              */     1    2
/*        2. UPDATES ALL TOTALS                                      */     1    2
/*  UNVPRNT IS THE MAIN ENTRY POINT WHICH PRINTS THE UNIV SUMMARY    */     1    2
/*  ************** DEFINITIONS OF IMPORTANT VARIABLES **************/     1    2
/*  SUBTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR A PARTICULAR      */     1    2
/*  AND ALL ASSOCIATED PCS                                           */     1    2
/*  SUBTOTAL2 : ACCUMULATES THE EXISTING NASF FOR A PARTICULAR ROOM- */     1    2
/*  TYPE AND ALL ASSOCIATED PCS                                      */     1    2
/*  GRANDTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR THE ENTIRE UNIV */     1    2
/*  GRANDTOTAL2 : ACCUMULATES THE EXISTING NASF FOR THE ENTIRE UNIV  */     1    2
/*  FLAG1 : SWITCH USED TO CONTROL OUTPUT PRINTOUT                   */     1    2
/*  ALL OTHER VARIABLES HAVE BEEN DEFINED IN THE MAIN PROCEDURE      */     1    2
/*                                                                   */     1    2
/*********************************************************************/     1    2
  DCL IUNIS, ACTIVITY_LOAD FIXED DEC(7,2), ROOM_CODE BIN FIXED,              1    2
  PROJNASF BIN FIXED(31), EXNASF BIN FIXED(31),NLOAD,PCS BIN FIXED;          1    2
        DCL LENUNIV(*),STPCSU(*,*) BIN FIXED, UENASF(*,*) BIN FIXED(31)      1    3
,       UALOAD(*,*,*) FIXED DEC(10,2), UPNASF(*,*,*) BIN FIXED(31), TOT     1    3
ALU(*,*) BIN FIXED(31), TABLE3(*,*) BIN FIXED, ROOM(*) CHAR(*),              1    3
        DESCRIP(*) CHAR(*);                                                  1    3
        DCL ( ROOMSUB (0:30)    CHAR(3), UNIT (0:30) CHAR(3)) EXTERNAL;      1    4
        DCL (NR,NC,NU,NL) EXTERNAL;                                          1    5
        DCL LOCATN CHAR(8) EXTERNAL, Q POINTER;                              1    6
        DCL DUMMY (1,1) BIN FIXED INIT(0), DEF BIN FIXED(31);                1    7
/*                                                                   */     1    8
       DCL  FLAG1  BIT(1) INIT('1'B), DEF  BIN FIXED,                        1    8
           (SUBTOTAL1,SUBTOTAL2) BIN FIXED(31) INIT (0),                     1    8
      (GRANDTOTAL1,GRANDTOTAL2) BIN FIXED (47) INIT (0);                     1    8
/*                                                                   */     1    9
        ON SUBSCRIPTRANGE                                                    1    9
           BEGIN;                                                            2   10
           IE=1;                                                             2   11
           LOCATN=' UNVPRNT';                                                2   12
           KFACTOR=88;                                                       2   13
           CALL ERRORM(Q,IE,KFACTOR,NLOAD,ICOL,IUNIV,IR,PCS,DUMMY,STPCSU     2   14
           );                                                                2   14
              IF IUNIS=0 THEN                                                3   15
              STOP;                                                          3   16
           PUT SKIP EDIT('HOWEVER THIS BEING A DEBUGGING RUN THE EXECUTI     2   17
           ON WILL BE FORCED')(A);                                           2   17
           END;                                                              2   18
        ON CONVERSION SYSTEM;                                                1   19
/*                                                                   */     1   20
        ON ENDPAGE(UNIVSUM)                                                  1   20
           BEGIN;                                                            2   21
           PUT PAGE FILE(UNIVSUM)EDIT('KANSAS STATE UNIVERSITY :   SPACE     2   22
            SUMMARY BY ROOM TYPES.')(COL(38),A);                             2   22
           PUT SKIP(3)FILE(UNIVSUM)EDIT('DESCRIPTION          ROOM-   PCS    2   23
            ACTIVITY UNIT  PROJECTED   EXISTING   SURPLUS/ % OF')(COL(24     2   23
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

          ),A)('TYPE','LOAD','NASF','NASF','DEFICIT','EXIS')(COL(43),A,   2   23
          COL(60),A,COL(76),A,COL(87),A,COL(94),A,COL(103),A);            2   23
          PUT SKIP(2)FILE(UNIVSUM);                                       2   24
          END;                                                            2   25
/*                                                                  */    1   26
/*                                                                  */    1   26
       SIGNAL ENDPAGE(UNIVSUM);                                           1   26
/*                                                                  */    2   27
RMU:      DO IR=1 TO NR;                                                 2   27
PCU:         DO IP=1 TO NU;                                              3   28
                IF STPCSU(IR,IP)=0 THEN                                   4   29
                GO TO EXTU;                                               4   30
                IF TOTALU(IR,IP)=0 THEN                                   4   31
                   IF UENASF(IR,IP)=0 THEN                                5   32
                   GO TO CONTPU;                                          5   33
             KOUNT=0;                                                     3   34
LFU:            DO IL=1 TO NL;                                            4   35
                JJ=TABLE3(IR,IL);                                         4   36
                   IF UPNASF(IR,IP,IL)=0 THEN                             5   37
                   GO TO CONTLU;                                          5   38
                KOUNT=KOUNT+1;                                            4   39
                   IF FLAG1 THEN                                          5   40
                      DO;                                                 6   41
                      FLAG1='0'B;                                         6   42
                      PUT SKIP(2)FILE(UNIVSUM)EDIT(DESCRIP(IR),ROOM(IR)   6   43
                      ||' '||ROOMSUB(JJ),STPCSU(IR,IP),UALOAD(IR,IP,IL)   6   43
                      ,UNIT(JJ),UPNASF(IR,IP,IL))(COL(21),A,COL(43),A,C   6   43
                      OL(51),F(3),COL(55),F(9,1),COL(66),A,COL(70),F(10   6   43
                      ));                                                 6   43
                      GO TO CONTLU;                                       6   44
                      END;                                                6   45
                PUT SKIP FILE(UNIVSUM)EDIT(ROOMSUB(JJ),STPCSU(IR,IP),UA   4   46
                LOAD(IR,IP,IL),UNIT(JJ),UPNASF(IR,IP,IL))(COL(47),A,COL   4   46
                (51),F(3),COL(55),F(9,1),COL(66),A,COL(70),F(10));        4   46
CONTLU:         END LFU;                                                  4   47
                IF FLAG1 THEN                                             4   48
                   DO;                                                    5   49
                   FLAG1='0'B;                                            5   50
                   PUT SKIP(2)FILE(UNIVSUM)EDIT(DESCRIP(IR),ROOM(IR),ST   5   51
                   PCSU(IR,IP),TOTALU(IR,IP),UENASF(IR,IP))(COL(21),A,C   5   51
                   OL(43),A,COL(51),F(3),COL(70),F(10),COL(81),F(10));    5   51
                   END;                                                   5   52
                ELSE                                                      4   53
                   DO;                                                    5   53
                      IF KOUNT=1 THEN                                     6   54
                      PUT SKIP(0)FILE(UNIVSUM)EDIT(UENASF(IR,IP))(COL(8   6   55
                      1),F(10));                                          6   55
                      ELSE                                                6   56
                      PUT SKIP FILE(UNIVSUM)EDIT('SUB-TOT',STPCSU(IR,IP   6   56
                      ),TOTALU(IR,IP),UENASF(IR,IP))(COL(43),A,COL(51),   6   56
                      F(3),COL(70),F(10),COL(81),F(10));                  6   56
                   END;                                                   5   57
                SUBTOTAL1=SUBTOTAL1+TOTALU(IR,IP);                        3   58
                SUBTOTAL2=SUBTOTAL2+UENASF(IR,IP);                        3   59
                GO TO CONTPU;                                             3   60
EXTU:           IF IP=1|IP=2 THEN                                         4   61
                GO TO JUMP;                                               4   62
             PUT SKIP FILE(UNIVSUM)EDIT('TOT',SUBTOTAL1,SUBTOTAL2)(COL(   3   63
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

            47),A,COL(70),F(10),COL(81),F(10));                                  3   63
JUMP:          IF SUBTOTAL2=0 THEN                                               4   64
               PUT SKIP(0)FILE(UNIVSUM)EDIT((SUBTOTAL2-SUBTOTAL1),'*')           4   65
               (COL(92),F(10),COL(103),A);                                       4   65
               ELSE                                                              4   66
                  DO;                                                            5   66
                  DEF=(SUBTOTAL2-SUBTOTAL1)*100/SUBTOTAL2;                       5   67
                  PUT SKIP(0)FILE(UNIVSUM)EDIT((SUBTOTAL2-SUBTOTAL1),D           5   68
                  EF)(COL(92),F(10),COL(103),F(4));                              5   68
                  END;                                                           5   69
            FLAG1='1'B;                                                          3   70
            GRANDTOTAL1=GRANDTOTAL1+SUBTOTAL1;                                   3   71
            GRANDTOTAL2=GRANDTOTAL2+SUBTOTAL2;                                   3   72
            SUBTOTAL1,SUBTOTAL2=0;                                               3   73
            GO TO CONTRU;                                                        3   74
CONTPU:     END PCU;                                                             3   75
CONTRU:  END RMU;                                                                2   76
         DEF=(GRANDTOTAL2-GRANDTOTAL1)*100/GRANDTOTAL2;                          1   77
         PUT SKIP(4)FILE(UNIVSUM)EDIT('GRAND-TOTAL',GRANDTOTAL1,GRANDTOTA        1   78
         L2,(GRANDTOTAL2-GRANDTOTAL1),DEF)(COL(30),A,COL(70),F(10),COL(81        1   78
         ),F(10),COL(92),F(10),COL(103),F(4));                                   1   78
         GO TO OVER;                                                             1   79
/*                                                                               */  1   80
/*                                                                               */  1   80
UNIVCHK:ENTRY(IUNIS,ACTIVITY_LOAD,ROOM_CODE,PROJNASF,EXNASF,NLOAD,PCS,0,         1   80
         LENUNIV,STPCSU,UENASF,UALOAD,UPNASF,TOTALU,TABLE3,ROOM,DESCRIP,L        1   80
         OAD_FACTOR);                                                            1   80
         IR=ROOM_CODE;                                                           1   81
            DO IUNIV=1 TO NU;                                                    2   82
               IF STPCSU(ROOM_CODE,IUNIV)=PCS THEN                               3   83
               GO TO TEST;                                                       3   84
               IF STPCSU(ROOM_CODE,IUNIV)=0 THEN                                 3   85
                  DO;                                                            4   86
                  LENUNIV(ROOM_CODE)=IUNIV;                                      4   87
                  STPCSU(ROOM_CODE,IUNIV)=PCS;                                   4   88
                  GO TO TEST;                                                    4   89
                  END;                                                           4   90
               IF STPCSU(ROOM_CODE,IUNIV)>PCS THEN                               3   91
                  DO;                                                            4   92
                  LENGTH=LENUNIV(ROOM_CODE);                                     4   93
                  NTIMES=LENGTH-IUNIV+1;                                         4   94
                     DO KCOL=1 TO NTIMES;                                        5   95
                     STPCSU(ROOM_CODE,LENGTH+1)=STPCSU(ROOM_CODE,LENGTH);        5   96
                     TOTALU(ROOM_CODE,LENGTH+1)=TOTALU(ROOM_CODE,LENGTH);        5   97
                     UENASF(ROOM_CODE,LENGTH+1)=UENASF(ROOM_CODE,LENGTH);        5   98
                     UALOAD(ROOM_CODE,LENGTH+1,*)=UALOAD(ROOM_CODE,LENGTH        5   99
                     ,*);                                                        5   99
                     UPNASF(ROOM_CODE,LENGTH+1,*)=UPNASF(ROOM_CODE,LENGTH        5  100
                     ,*);                                                        5  100
                     LENGTH=LENGTH-1;                                            5  101
                     END;                                                        5  102
                  LENUNIV(ROOM_CODE)=LENUNIV(ROOM_CODE)+1;                       4  103
                  STPCSU(ROOM_CODE,IUNIV)=PCS;                                   4  104
                  UENASF(ROOM_CODE,IUNIV),TOTALU(ROOM_CODE,IUNIV)=0;             4  105
                  UALOAD(ROOM_CODE,IUNIV,*)=0.;                                  4  106
                  UPNASF(ROOM_CODE,IUNIV,*)=0.;                                  4  107
                  GO TO TEST;                                                    4  108
                  END;                                                           4  109
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

            END;                                                          2  110
TEST:       IF LOAD_FACTOR=0 THEN                                         2  111
            PROJNASF=EXNASF;                                              2  112
         IN=ROOM_CODE;                                                    1  113
         UALOAD(IN,IUNIV,NLOAD)=UALOAD(IN,IUNIV,NLOAD)+ACTIVITY_LOAD;     1  114
         UPNASF(IN,IUNIV,NLOAD)=UPNASF(IN,IUNIV,NLOAD)+PROJNASF;          1  115
         IUNIS=IUNIV;                                                     1  116
OVER:    END UNVPRNT;                                                     1  117
```

Figure A.22. Flow Chart of PCSPRNT

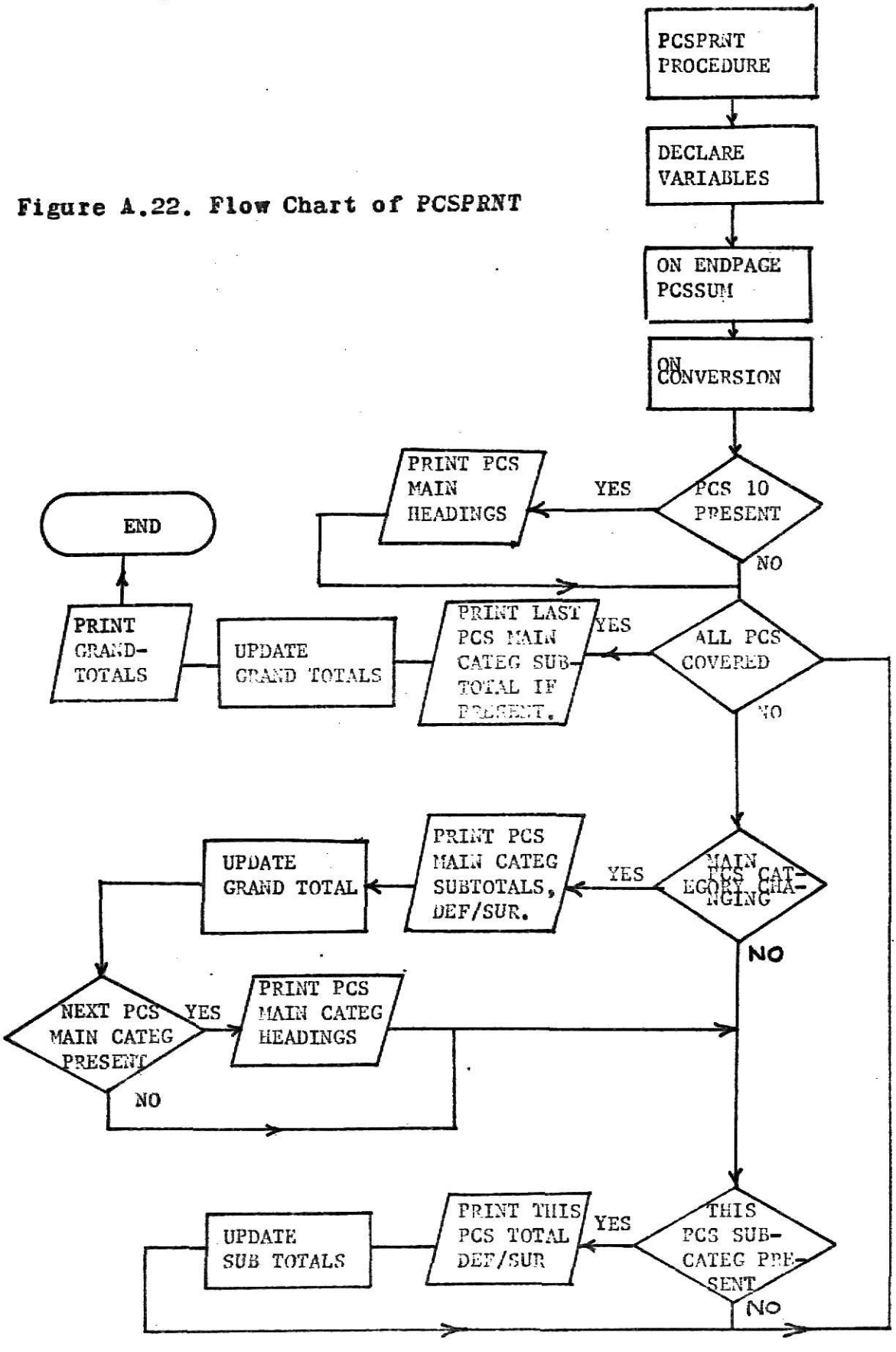


Figure A.23. PCSPRNT: Source Listings

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

(NOSUBSCRIPTRANGE):PCSPRINT:                                                    1    1
        PROCEDURE(PCSSWITCH);                                                   1    1
/*************************************************************************/    1    2
/*                                                                       */    1    2
/*  THIS PROCEDURE PRINTS OUT THE PROGRAM CLASSIFICATION SUMMARY         */    1    2
/*  ..................... DEFINITIONS OF IMPORTANT VARIABLES .......     */    1    2
/*  SUBTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR EACH PCS MAIN-        */    1    2
/*  CATEGORY                                                             */    1    2
/*  SUBTOTAL2 : ACCUMULATES THE EXISTING NASF FOR EACH PCS MAIN -        */    1    2
/*  CATEGORY                                                             */    1    2
/*  GRANDTOTAL1 : ACCUMULATES THE PROJECTED NASF FOR THE ENTIRE UNIV     */    1    2
/*  GRANDTOTAL2 : ACCUMULATES THE EXISTING NASF FOR THE ENTIRE UNIV      */    1    2
/*  ALL OTHER VARIABLES HAVE BEEN DEFINED IN THE MAIN PROCEDURE          */    1    2
/*  ..................... ON CONDITION UNITS .........................    */    1    2
/*                                                                       */    1    2
/*************************************************************************/    1    2
                                                                  DCL P C       1    2
SSWITCH (*) BIT(1);                                                             1    2
        DCL ( PCSVAL(40)  FIXED BIN,   PPNASF(40) BIN FIXED(31) ,               1    3
          PENASF(40) BIN FIXED(31)) EXTERNAL, PCSDEF(8) CHAR(20) EXTERN         1    3
AL;                                                                             1    3
        DCL PCSSUBDEF(40) CHAR(20), STKK FIXED BIN;                             1    4
        DCL (SUBTOTAL1,SUBTOTAL2) BIN FIXED(31) INIT (0),                       1    5
        (GRANDTOTAL1,GRANDTOTAL2) BIN FIXED (47)  INIT (0), DEF BIN FIX         1    5
ED (31),    STPCSPNASF FIXED BIN(31) INIT (0);                                  1    5
/*                                                                   */        1    6
/*                                                                   */        1    6
        ON ENDPAGE(PCSSUM)                                                      1    6
           BEGIN;                                                               2    7
           PUT PAGE FILE(PCSSUM)EDIT('KANSAS STATE UNIVERSITY :   SPACE         2    8
           SUMMARY BY PCS')(COL(34),A);                                         2    8
           PUT SKIP(2)FILE(PCSSUM)EDIT('PCS     . DESCRIPTION          PRO      2    9
           JECTED    EXISTING    SURPLUS/  % OF')(COL(33),A)('NASF             2    9
             NASF    DEFECIT   PROJ')(SKIP,COL(67),A);                         2    9
           PUT SKIP(2)FILE(PCSSUM);                                             2   10
           END;                                                                 2   11
        ON CONVERSION SYSTEM;                                                   1   12
FORMAT1:FORMAT(COL(5),A,COL(61),P'(10)Z',COL(74),P'(8)Z(2)9',COL(85),F(         1   13
        10),COL(97),F(4));                                                      1   13
FORMAT2:FORMAT(COL(5),A,COL(61),P'(10)Z',COL(74),P'(8)Z(2)9',COL(85),F(         1   14
        10),COL(100),A);                                                        1   14
        SIGNAL ENDPAGE(PCSSUM);                                                 1   15
        GET SKIP FILE(SYSIN)EDIT((PCSSUBDEF(JJ)DO JJ=1 TO 40))(A(20));          1   16
        STKK=10;                                                                1   17
        II=1;                                                                   1   18
           IF PCSSWITCH(II)THEN                                                 2   19
           PUT SKIP(2)FILE(PCSSUM)EDIT(STKK,PCSDEF(II))(COL(33),F(2),COL        2   20
           (37),A);                                                             2   20
PCSLOOP:   DO JJ=1 TO 40;                                                       2   21
           KK=PCSVAL(JJ)/10;                                                    2   22
           KK=KK*10;                                                            2   23
           II=KK/10;                                                            2   24
              IF STKK¬=KK THEN                                                  3   25
                 DO;                                                            4   26
                    IF SUBTOTAL2=0 THEN                                         5   27
                       IF SUBTOTAL1=0 THEN                                      6   28
                       GO TO NEXT;                                              6   29
                       ELSE                                                     6   30
```

```
KSU'S PL/I NEATENER AND PRECOMPILER                    PAGE

                        IF SUBTOTAL1=STPCSPNASF THEN                    7   30
                        GO TO NEXT;                                     7   31
                    IF SUBTOTAL2=0 THEN                                 5   32
                    PUT SKIP FILE(PCSSUM)EDIT('SUB-TOTAL',SUBTOTAL1,SUBT  5   33
                    OTAL2,(SUBTOTAL2-SUBTOTAL1),'*')(R(FORMAT2));       5   33
                    ELSE                                                5   34
                        DO;                                             6   34
                        DEF=(SUBTOTAL2-SUBTOTAL1)*100/SUBTOTAL2;        6   35
                        PUT SKIP FILE(PCSSUM)EDIT('SUB-TOTAL',SUBTOTAL1,S  6   36
                        UBTOTAL2,(SUBTOTAL2-SUBTOTAL1),DEF)(R(FORMAT1));   6   36
                        END;                                            6   37
NEXT:               GRANDTOTAL1=GRANDTOTAL1+SUBTOTAL1;                  4   38
                    GRANDTOTAL2=GRANDTOTAL2+SUBTOTAL2;                  4   39
                    SUBTOTAL1,SUBTOTAL2=0;                              4   40
                    STKK=KK;                                            4   41
                        IF PCSSWITCH(II)THEN                            5   42
                        PUT SKIP(2)FILE(PCSSUM)EDIT(KK,PCSDEF(II))(COL(33),F  5   43
                        (2),COL(37),A);                                 5   43
                    END;                                                4   44
                IF PENASF(JJ)=0 THEN                                    3   45
                    IF PPNASF(JJ)=0 THEN                                4   46
                    GO TO EPLOOP;                                       4   47
                IF PENASF(JJ)=0 THEN                                    3   48
                PUT SKIP FILE(PCSSUM)EDIT(PCSVAL(JJ),PCSSUBDEF(JJ),PPNASF(  3   49
                JJ),PENASF(JJ),(PENASF(JJ)-PPNASF(JJ)),'*')(COL(35),F(2),C  3   49
                OL(38),A,COL(61),P'(10)Z',COL(74),P'(10)Z',COL(85),F(10),C  3   49
                OL(100),A);                                             3   49
                ELSE                                                    3   50
                    DO;                                                 4   50
                    DEF=(PENASF(JJ)-PPNASF(JJ))*100/PENASF(JJ);         4   51
                    PUT SKIP FILE(PCSSUM)EDIT(PCSVAL(JJ),PCSSUBDEF(JJ),PPNA  4   52
                    SF(JJ),PENASF(JJ),(PENASF(JJ)-PPNASF(JJ)),DEF)(COL(35),  4   52
                    F(2),COL(38),A,COL(61),P'(10)Z',COL(74),P'(10)Z',COL(85  4   52
                    ),F(10),COL(97),F(4));                              4   52
                    END;                                                4   53
            SUBTOTAL1=SUBTOTAL1+PPNASF(JJ);                             2   54
            SUBTOTAL2=SUBTOTAL2+PENASF(JJ);                             2   55
            STPCSPNASF=PPNASF(JJ);                                      2   56
EPLOOP:     END PCSLOOP;                                                2   57
            IF SUBTOTAL2=0 THEN                                         2   58
                IF SUBTOTAL1=0 THEN                                     3   59
                GO TO EPCSPRNT;                                         3   60
                ELSE                                                    3   61
                    IF SUBTOTAL1=STPCSPNASF THEN                        4   61
                    GO TO EPCSPRNT;                                     4   62
            IF SUBTOTAL2=0 THEN                                         2   63
            PUT SKIP FILE(PCSSUM)EDIT('SUB-TOTAL',SUBTOTAL1,SUBTOTAL2,(SU  2   64
            BTOTAL2-SUBTOTAL1),'*')(R(FORMAT2));                        2   64
            ELSE                                                        2   65
                DO;                                                     3   65
                DEF=(SUBTOTAL2-SUBTOTAL1)*100/SUBTOTAL2;                3   66
                PUT SKIP FILE(PCSSUM)EDIT('SUB-TOTAL',SUBTOTAL1,SUBTOTAL2,  3   67
                (SUBTOTAL2-SUBTOTAL1),DEF)(R(FORMAT1));                 3   67
                END;                                                    3   68
EPCSPRNT:                                                               1   69
        GRANDTOTAL1=GRANDTOTAL1+SUBTOTAL1;                              1   69
        GRANDTOTAL2=GRANDTOTAL2+SUBTOTAL2;                              1   70
        DEF=(GRANDTOTAL2-GRANDTOTAL1)*100/GRANDTOTAL2;                  1   71
```

```
KSU'S PL/I BEAUTIFIER AND PRECOMPILER                    PAGE

        PUT SKIP(4)FILE(PCSSUMMEDIT)('     TOTAL',GRANDTOTAL1,GRANDTOTAL2,   1   72
        (GRANDTOTAL2-GRANDTOTAL1),DEF)(R(FORMAT1));                          1   72
        END PCSPRNT;                                                         1   73
```

```
KSU'S PL/I ...ATENER AND PRECOMPILER                    PAGE

ERRORM: PROCEDURE(Q,ERROR_CODE,IFACTOR,ILOAD,JCOL,JUNIV,R_CODE,PC,STPCSC  1     1
        ,STPCSU);                                                         1     1
/*                                                                      */  1     2
/*  ********************************************************************/  1     2
/*  THIS PROCEDURE IS CALLED BY THE OTHER PROCEDURES WHENEVER AN      */  1     2
/*  ERROR CONDITION IS ENCOUNTERED. APPROPRIATE DIAGNOSTICS ARE       */  1     2
/*  GIVEN                                                             */  1     2
/*                                                                    */  1     2
/*  ********************************************************************/  1     2
 DCL (ERROR_CODE,IFACTOR,ILOAD,JCOL,JUNIV,R_CODE,PC) BIN FIXED;           1     2
        DCL Q POINTER,  CARD1 CHAR(80) BASED (Q);                         1     3
        DCL PCS BIN FIXED,  LOCATN CHAR(8) EXTERNAL, ROOM_CODE BIN FIXE   1     4
0;                                                                        1     4
        DCL STPCSC(*,*) BIN FIXED, STPCSU(*,*) BIN FIXED;                 1     5
        DCL (NR,NL,NC,NU) EXTERNAL;                                       1     6
        DCL ACTION (3) LABEL;                                             1     7
        PCS=PC;                                                           1     8
        NLOAD=ILOAD;                                                      1     9
        ICOL=JCOL;                                                        1    10
        IUNIV=JUNIV;                                                      1    11
        LOAD_FACTOR=IFACTOR;                                              1    12
        ROOM_CODE=R_CODE;                                                 1    13
        GO TO ACTION(ERROR_CODE);                                         1    14
ACTION(1):                                                                1    15
        PUT SKIP(4)EDIT('SUBSCRIPTRANGE CONDITION HAS BEEN RAISED IN PRO  1    15
        CEDURE : ',LOCATN)(COL(5),A,A)('ECHO CHECK OF INPUT :  ',CARD1)(  1    15
        SKIP(2),COL(5),A,SKIP(2),COL(5),A);                               1    15
        PUT SKIP DATA(LOAD_FACTOR,NLOAD,ICOL,IUNIV);                      1    16
        PUT SKIP EDIT(' THE STORED PCS VALUES IN THE COLLEGE & UNIVERSIT  1    17
        Y ARRAYS WERE : ')(A);                                            1    17
           IF LOCATN=' COLPRNT' THEN                                      2    18
           PUT SKIP EDIT(STPCSC(ROOM_CODE,*))(SKIP,X(1),F(2));            2    19
           ELSE                                                           2    20
              IF LOCATN=' UNVPRNT' THEN                                   3    20
              PUT SKIP EDIT(STPCSU(ROOM_CODE,*))(SKIP,X(1),F(2));         3    21
              ELSE                                                        3    22
              PUT SKIP EDIT('NOT RELEVANT')(A);                           3    22
        PUT SKIP(3)EDIT('.....CHECK THE DIMENSION STATEMENTS..... TH      1    23
        E EXECUTION WILL BE TERMINATED')(COL(5),A);                       1    23
        RETURN;                                                           1    24
ACTION(2):                                                                1    25
        PUT SKIP(4)EDIT('DATA INTERRUPT.... CONVERSION..... ECHO CHECK O  1    25
        F INPUT :  ',CARD1)(COL(5),A,A)('THIS DATA WILL BE IGNORED AND T  1    25
        HE AREAS WILL NOT ENTER THE SUMMARIES')(SKIP,COL(5),A);           1    25
        RETURN;                                                           1    26
ACTION(3):                                                                1    27
        PUT SKIP(4)EDIT('ERROR.......CARDS OUT OF SEQUENCE ; ECHO CHECK   1    27
        OF INPUT : ',CARD1)(COL(5),A)('..... EXECUTION WILL BE TERMIN     1    27
        ATED')(SKIP(4),COL(5),A);                                         1    27
        RETURN;                                                           1    28
ACTION(4):                                                                1    29
        PUT SKIP(4)FILE(SYSPRINT)EDIT(' INPUT ERROR ..... IMPROPER PCS    1    29
         ..... ECHO CHECK OF INPUT:  ',CARD1)(COL(5),A,SKIP,COL(5),       1    29
        A);                                                               1    29
        PUT SKIP DATA(ROOM_CODE,LOAD_FACTOR,PCS);                         1    30
        PUT SKIP EDIT('THIS DATA WILL BE IGNORED AND THE AREAS WILL NOT   1    31
        ENTER THE SUMMARIES')(COL(5),A);                                  1    31
        RETURN;                                                           1    32
```

Figure A.24. ERRORM: Source Listings

```
KSU'S PL/I WATFIVER AND PRECOMPILER                    PAGE

ACTION(5):                                                              1    33
        PUT SKIP(4)FILE(SYSPRINT)EDIT(' INPUT ERROR ..... IMPROPER LOA  1    33
        D FACTOR..... ECHO CHECK OF INPUT : ',CARD1)(COL(5),A,SKIP,COL(  1    33
        5),A);                                                          1    33
        PUT SKIP DATA(ROOM_CODE,LOAD_FACTOR,NLOAD);                     1    34
        PUT SKIP EDIT('THIS DATA WILL BE IGNORED AND THE AREAS WILL NOT  1    35
        ENTER THE SUMMARIES')(COL(5),A);                                1    35
        RETURN;                                                         1    36
ACTION(6):                                                              1    37
        PUT SKIP(4)EDIT('KEY CONDITION RAISED FOR  THE DEMAND FILE ..... 1    37
        ..... ECHO CHECK OF INPUT : ',CARD1)(COL(5),A,A)('THE INPUT     1    37
        INFORMATION FOR THE ABOVE DEPARTMENT WILL BE IGNORED')(SKIP(3),C 1    37
        OL(5),A);                                                       1    37
        ISKIP=LOAD_FACTOR-1;                                            1    38
        READ FILE(INPUT1)IGNORE(ISKIP);                                 1    39
        RETURN;                                                         1    40
ACTION(7):                                                              1    41
        PUT SKIP(4)EDIT('KEY CONDITION RAISED FOR  THE SPACED FILE ..... 1    41
        ..... ECHO CHECK OF INPUT : ',CARD1)(COL(5),A,A)('THE INPUT     1    41
        INFORMATION FOR THE ABOVE DEPARTMENT WILL BE IGNORED')(SKIP(3),C 1    41
        OL(5),A);                                                       1    41
        ISKIP=LOAD_FACTOR-1;                                            1    42
        READ FILE(INPUT1)IGNORE(ISKIP);                                 1    43
        RETURN;                                                         1    44
ACTION(8):                                                              1    45
        PUT SKIP(4)EDIT('CONVERSION ERROR.............. IMPROPER  DEPARTM 1    45
        ENT CODE..... ECHO CHECK OF INPUT :  ',CARD1)(COL(5),A,A)('T     1    45
        HE EXECUTION WILL BE TERMINATED')(SKIP(3),COL(5),A);            1    45
        RETURN;                                                         1    46
        END ERRORM;                                                     1    47
```

Figire A.24. ERRORM: Source Listings (continued)

```
KSU'S PL/I CREATENER AND PRECOMPILER                    PAGE

TITLES: PROCEDURE(FLAG4,FLAG5,FLAG6);                                    1    1
/*                                                                      */  1    2
/*  THIS PROCEDURE PRINTS NECESSARY INFORMATION ON THE TITLE PAGE       */  1    2
/*                                                                      */  1    2
/*  ********************************************************************/  1    2
    DCL (FLAG4,FLAG5,FLAG6) BIT(1);                                       1    2
        OPEN FILE(HEAD)PRINT LINESIZE(132);                               1    3
        PUT SKIP(3)FILE(HEAD)EDIT('UNIVERSITY  SPACE  PLANNING')(COL(45   1    4
        ),A)('PROJECTIONS FOR KANSAS STATE UNIV.')(SKIP(3),COL(54),A)('V  1    4
        ERSION I  (JAN 1973)')(SKIP(4),COL(52),A)('AUTHORS :  DR. L.E.GR  1    4
        OSH &')(SKIP(4),COL(45),A)('A.CHANDRASHEKAR')(SKIP,COL(55),A)('D  1    4
        EPT OF INDUSTRIAL ENGG.')(SKIP(3),COL(55),A)('KANSAS STATE UNIVE  1    4
        RSITY.')(SKIP,COL(55),A);                                         1    4
        PUT SKIP(4)FILE(HEAD)EDIT('THE FOLLOWING SUMMARIES HAVE BEEN REQ  1    5
        UESTED :')(COL(50),A)('1.DEPARTMENT - ON ROOM TYPE MAJOR & PCS MIN 1    5
        OR  (DEFAULT)')(SKIP(2),COL(55),A);                               1    5
           IF FLAG4 THEN                                                  2    6
           PUT SKIP(2)FILE(HEAD)EDIT('COLLEGE - ON ROOM TYPE MAJOR & PCS  2    7
            MINOR (I), ROOM TYPE MAJOR (II)')(COL(55),A);                 2    7
           IF FLAG5 THEN                                                  2    8
           PUT SKIP(2)FILE(HEAD)EDIT('UNIVERSITY - ON ROOM TYPE MAJOR &   2    9
           PCS MINOR')(COL(55),A);                                        2    9
           IF FLAG6 THEN                                                  2   10
           PUT SKIP(2)FILE(HEAD)EDIT('SUMMARY BY PROGRAM CLASSIFICATION   2   11
           STRUCTURE')(COL(55),A);                                        2   11
        END TITLES;                                                       1   12
```

Figure A.25. TITLES: Source Listings

APPENDIX B

USER MANUAL

The purpose of this section is to acquaint the user with all the necessary input details, job control, etc. There are three input card files. These are:

i) FILE SYSIN: The following details are read in from this file:

   a. Room Type definitions

   b. Room Type codes

   c. Units of the Activity Load

   d. Activity Load descriptions, e.g. FAC (faculty, JS (junior-senior), etc.

   e. PCS definitions

   f. PCS values.

   The above details are not subject to frequent changes.

ii) FILE CONTROL: The options which control the logic of the main routine are read in from this file. They are:

   a. The program furnishes the college, university, and PCS summaries only if needed. The value of the variables COL_SUMMARY, UNIV_SUMMARY, and PCS_SUMMARY decides if the respective summary is required or not. If the value is 'NO', then the summary is not given. Also the value of the switch ISTOP is read in from the same card. If the value of this switch is zero, then the execution is terminated whenever a subscriptrange error is detected.

The format of the first card is as follows:

Columns 1-4 - COL_SUMMARY : punch in 'NO' right justified if this summary is not required.

Columns 5-8 - UNIV_SUMMARY : punch in 'NO' right justified if this summary is not required.

Columns 9-12 - PCS_SUMMARY : punch in 'NO' right justified if this summary is not required.

Columns 13-16 - ISTOP : Punch in any number other than zero, if you want the execution to proceed when any severe error is detected.

b. The value of the two dimensioned variable TABLE3 is read in from this file. TABLE3 is a 13x15 matrix. It has been assumed that each room type will have at the maximum five sub-categories. TABLE3 stores the value of the load factors associated with each of the room types. TABLE3 is shown in Figure B.1. The use of the matrix can be better explained by taking a concrete example. Let us consider Room Type 200 (Room Code =2). The sub-categories of this room type are:

   a. Freshman-Sophomore (FS) - Load Factor = 2,
   b. Junior-Senior (JS) - Load Factor = 3,
   c. Beginning Graduates (G1) - Load Factor = 4,
   d. Advanced Graduates (G2) - Load Factor = 5,
   e. No data available, or function not existing - Load Factor = 0.

These load factor values are stored in the second row of this table, the room code indicating the row number. The main

					ROOM_TYPE
1	0	0	0	0	100
2	3	4	5	0	200
11	12	13	14	0	250
15	16	17	12	0	300
21	10	0	0	0	410
22	0	0	0	0	420
0	0	0	0	0	440
0	0	0	0	0	500
0	0	0	0	0	600
0	0	0	0	0	700
0	0	0	0	0	800
23	24	0	0	0	900
0	0	0	0	0	888

Figure B.1. Table3

routine searches the second row of this table for the load factor value read in, and returns the column number containing this load-factor. This number is the value of the third subscript in the three dimensioned arrays- CALOAD, CPNASF, UALOAD and UPNASF. While printing out the college and university summaries, the value of the third subscript of these arrays is used to obtain the appropriate load factor value from this table. The 65 values of this TABLE3 are punched with Format F(4).

iii) FILE INPUT1: This is a card file sequenced on department code. Each record is 80 bytes. Figure B.2 shows the card format. The following information is read in from this file:

A. FOR EACH DEPARTMENT:

a. Department Code - 4 bytes.

b. Card number - 1 byte.

c. Total cards for this department - 1 byte.

d. ROOM_CODE, LOAD_FACTOR,SPACE_FACTOR and PCS-totally 6 fields, each 11 bytes, separated by a blank.

e. There is a facility to read in activity load data from this file. A '99' in the load factor field is an indicator that the activity load data is to be read in from this file. The exact value of the load factor and the corresponding activity load is read in from the 11 byte field immediately following.

f. A '99' in the Room_Code field is a signal to skip that particular 11 bytes field.

g. Room_Code = '0' i.e. a blank 11 byte field signifies 'END OF DATA' for that particular department.

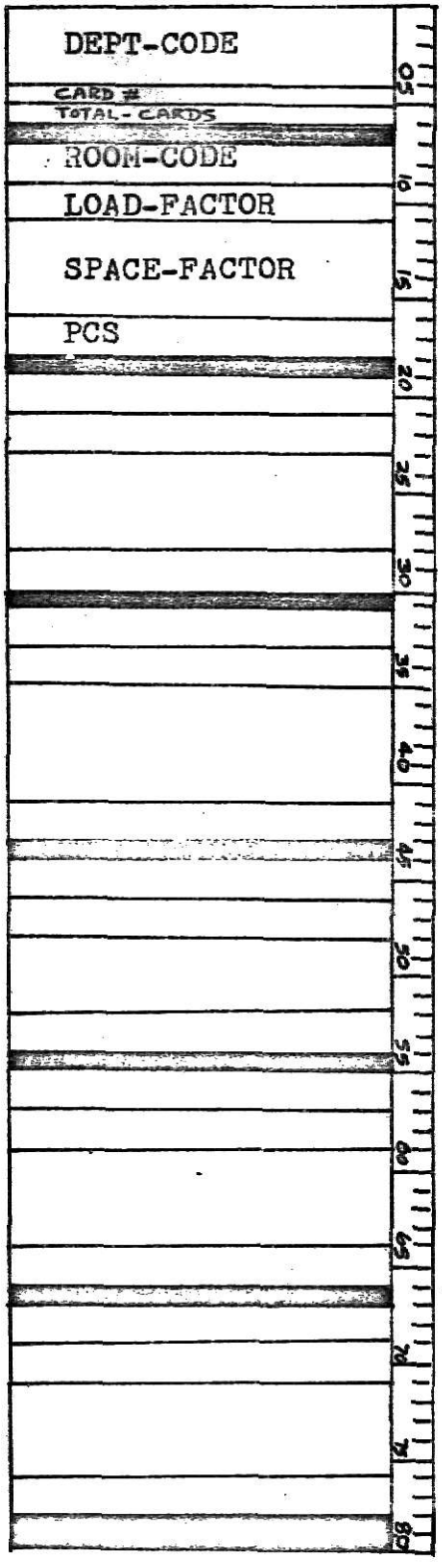


Figure B.2. Card Format of File INPUT1

B. FOR EACH COLLEGE:

a. It was not possible to develop a truly sequential set of department codes for the reasons explained in Section 4.3. The information contained in the department code was not sufficient to recognize the college affiliation. To get around this difficulty an alternate method was developed. 'NEW' punched in the first four columns is a signal indicating that input information for a new college is commencing. From the same card the following details are read in:

Bytes 5-8: College Administration Code

Bytes 9-12: College Hegis Code

Bytes 13-32: College Name.

b. 'OVER' punched in the first four columns is a signal indicating that the input information for the previous college is over. Procedure COLPRNT is called to print college totals and initialize storage areas.

JOB CONTROL NEEDED TO USE THE PROGRAM.

WHEN MAKING A RUN WITH SOURCE DECKS.

```
//          JOB (FH08R273,9,5),SHEKAR,CLASS=D
//          EXEC PL1LFCLG
//PL1L.SYSIN DD *
                    'SPPROJ' source deck
*PROCESS
                    'ERRORM' source deck
*PROCESS
                    'TITLES' source deck
```

(cont.)

```
*PROCESS
                    'COLPRNT' source deck
*PROCESS
                    'UNVPRNT' source deck
*PROCESS
                    'PCSPRNT' source deck
/*
//GO.SYSIN DD *
                a.  Room Type Description
                b.  Room Type Codes
                c.  Load Factor Units
                d.  Room Type Sub-Category Definitions
                e.  PCS Definitions (main)
                f.  PCS Codes
                g.  PCS Definitions (sub-categories)
/*
//GO.CONTROL DD *
                a.  Value of COL_SUMMARY, UNIV_SUMMARY, PCS_SUMMARY, ISTOP
                b.  TABLE3
/*
//GO.INPUT1 DD  *
                Input Cards
/*
//GO.DEMANDD DD DSN=COFH08.DEMAND,UNIT=2314,
//          VOL=SER=111111,SPACE=(91,150),DISP=(OLD,KEEP),
//          DCB=(RECFM=F,BLKSIZE=91,DSORG=DA,KEYLEN=4)
```

(cont.)

```
//GO.SPACED DD DSN=COFHO8.SPACEX,UNIT=2314,
//          VOL=SER=111111,SPACE=(120,150),DISP=(OLD,KEEP),
//          DCB=(RECFM=F,BLKSIZE=120,DSORG=DA,KEYLEN=4)
//GO.HEAD DD SYSOUT=A
//GO.DEPSUM DD SYSOUT=A
//GO.COLSUMA DD SYSOUT=A
//GO.COLSUMB DD SYSOUT=A
//GO.UNIVSUM DD SYSOUT=A
//GO.PCSSUM DD SYSOUT=A
/* 360/50
```

Note: Procedures SPPROJ, ERRORM & TITLES should be present always. Procedures COLPRNT, UNVPRNT and PCSPRNT need be included only if the respective summaries are required.

MAKING A RUN WITH OBJECT DECKS

```
//          JOB (FH08R273,5,5),SHEKAR,CLASS=D
//          EXEC PL1LFLG
//LKED.SYSIN DD *
                  'SPPROJ' object deck
                  'TITLES' object deck
                  'ERRORM' object deck
                  'COLPRNT' object deck
                  'UNVPRNT' object deck
                  'PCSPRNT' object deck
ENTRY IHENTRY
/*
```

GO STEP CARDS ARE THE SAME

NOTE: Object decks for procedures SPPROJ, ERRORM and TITLES should be present always. Object decks for procedures COLPRNT, UNVPRNT and PCSPRNT need be included only if the respective summaries are required.

EXECUTING A LOAD MODULE

Let us assume these procedures were compiled and link edited seperately (with linkage editor option NCAL being specified), the load module being saved in the catalogued data set COFHO8.SHEKAR.LOADLIB. The job control required to execute these load modules will be:

```
//          JOB (FH08R273,6,5),SHEKAR,CLASS=D
//          EXEC PL1LFLG
//LKED.SYSIN DD *
 ENTRY IHENTRY
 INCLUDE DD1(SPROJ,TITLES,ERRORM,COLPRNT,UNVPRNT,PCSPRNT)
/*
//LKED.DD1 DD DSN=COFHO8.SHEKAR.LOADLIB

        GO STEP CARDS ARE THE SAME
```

NOTE: In the INCLUDE card, procedures SPPROJ, TITLES and ERRORM should always be present. COLPRNT, UNVPRNT and PCSPRNT need be included only if the respective summaries are required.

APPENDIX C

SOME SPECIAL FEATURES AVAILABLE IN PL/I

C.1 FILE HANDLING IN PL/I.

This section will present a brief discussion on the different keyed files available in PL/I, and the relative advantages and disadvantages of these organizations. For additional details the readers are referred to the IBM PL/I Language Reference Manual and the Programmer's Guide. The material presented here has been borrowed in part or whole from these manuals.

RECORD KEYS: There are two different kinds of keys, recorded keys and source keys. A recorded key is a character string that immediately precedes each record in the data set to identify that record; its length cannot exceed 255 characters. A source key is the character string value of the expression that appears in the KEY or KEYFROM option of the data trnasmission to identify the record to which the statement refers.

INDEXED SEQUENTIAL FILE: Records in this file are in logical sequence according to the keys that are associated with each record. The key is a character string that usually represents an item within the record. Logical records are arranged in the data set in ascending key sequence according to the system/360 collating sequence. V_Format and F_Format records, blocked or unblocked, can be used in an Indexed Sequential Data.

Indexed Sequential data set have to be created sequentially; but, once created, the associated file may have the attributes SEQUENTIAL or DIRECT as well as INPUT or UPDATE. When the file has DIRECT attribute, records may be retrieved, added, deleted, and replaced at random.

However this organization has one major drawback. The space allotted to this data set is in terms of cylinders, the minimum allocation being 1 cylinder (20 tracks on the 2314 unit). The Existing Space Inventory file, and the Demand Data file developed for the space planning work, require only about 5 tracks at the maximum. Hence Indexed Sequential Organization was not chosen, because it leads to poor utilization of direct access storage. However Indexed Sequential files are relatively easier to work with, when compared to the other direct organizations.

REGIONAL ORGANIZATION: A data set with Regional Organization is divided into regions, each of which is identified by a region number, and each of which may contain one record or more than one record, depending on the type of Regional Organization. The regions are numbered in succession, beginning with zero, and a record is accessed by specifying its region number, and perhaps a key, in data transmission statement (record key). The length of the recorded keys in a REGIONAL data set is defined by the KEYLEN subparameter of the DD statement that defines the data set.

A REGIONAL data set can be created in a similar manner to an INDEXED data set, records being presented in the order of ascending region numbers; alternatively, direct access can be used, in which case the records can be presented in random sequence. Once a REGIONAL data set has been created, it can be accessed by a file with the attributes SEQUENTIAL or DIRECT as well as INPUT or UPDATE. When the file has the DIRECT attribute, records can be retrieved, added, deleted, and replaced at random.

TYPES OF REGIONAL ORGANIZATION: There are three types of REGIONAL organization. These are:

REGIONAL(1) ORGANIZATION: A REGIONAL(1) data set contains unblocked F-Format

records that do not have recorded keys. Each region in the data set contains only one record; therefore each region number corresponds with a relative record position within the data set. However for the application on hand, since there are recorded keys, this organization is not recommended. It takes up too much direct access storage.

REGIONAL(2) ORGANIZATION: A REGIONAL(2) data set contains unblocked F-Format records that have recorded keys. Each region in the data set contains only one record. Direct access of REGIONAL(2) data set employs the region number specified in a source key to locate the required region. Once the region has been located, a sequential search for space to add a new record, can be made. However the author encountered a lot of problems in creating this data set. It is not clear if each region number corresponds to a seperate track or not. Apparently it does.

REGIONAL(3) ORGANIZATION: A REGIONAL(3) data set contains unblocked F-Format, V-Format, or U-Format records with recorded keys. Each region in the data set corresponds with a track on a direct access device, and can contain one or more records. Direct access of REGIONAL(3) data set is similar to that for a REGIONAL(2) data set. This organization was found ideal for the application on hand. The Department Codes serve as the Record Keys. A simple hash code was set up, which computed the region numbers. The region numbers ranged from 0 to 4, signifying a maximum of 5 regions (or tracks) needed for the data set.

C.2 STRUCTURES IN PL/I.

(Ref: A Guide to PL/I by S. V. Pollack and T. D. Sterling)

Although arrays represnet a convenient and flexible vehicle for organizing and handling collections of data, there is a basic restriction

connected with their use. Successful construction of an array must meet a criterion for consistency; all its members must possess a common set of attributes. But it is often desirable to assign a collective name to a group of data that may or may not necessarily share a set of common attributes, but are nonetheless related to each other. Each record in our Existing Space Inventory file (say) is a good example for this. The name 'EXSPDATA' refers to information which have different attributes. The Hegis, Department Code, and Department Name are character strings, while the space data (SPDATA) has fourteen combinations of Room Type (Bin Fixed), PCS (Bin Fixed), and Existing Space (Bin Fixed(31)).

The name appearing in the Declare is the structure name. A structure is declared by listing the major structure name, followed by a list of names to be subservient to that structure. The relationship between parts of a structure is indicated by positive integer constants called levels placed before each name and separated from it by one or more blanks. A major structure name is always coded as level 1. The level numbers given to minor structures and elements depend on their placement and status in the structural tree.

C.3 EXTERNAL PROCEDURES IN PL/I.

External procedures in PL/I is similar to the subroutine in FORTRAN. External procedures are independent entity, and can be compiled and link edited separately. For Programming facility and efficiency, it is a good practice to split up the entire programming work into separate modules (procedures) and debug them separately; and finally when the individual modules are functioning properly, they can be linked together and executed.

APPENDIX D

HEGIS TAXONOMY CODES

Ref: National Center for Educational Statistics

THE HEGIS TAXONOMY*

The reader should note that a few modifications have been made to adapt this list to use for facilities inventory. See Education 0800 and General 8000.

0100 AGRICULTURE AND NATURAL RESOURCES
- 0101 Agriculture, General
- 0102 Agronomy (Field Crops, and Crop Management)
- 0103 Soils Science (Management and Conservation)
- 0104 Animal Science (Husbandry)
- 0105 Dairy Science (Husbandry)
- 0106 Poultry Science
- 0107 Fish, Game, and Wildlife Management
- 0108 Horticulture (Fruit and Vegetable Production)
- 0109 Ornamental Horticulture (Floriculture, Nursery Science)
- 0110 Agricultural and Farm Management
- 0111 Agricultural Economics
- 0112 Agricultural Business
- 0113 Food Science and Technology
- 0114 Forestry
- 0115 Natural Resources Management
- 0116 Agriculture and Forestry Technologies
- 0117 Range Management
- 0199 Other, Specify

0200 ARCHITECTURE AND ENVIRONMENTAL DESIGN
- 0201 Environmental Design, General
- 0202 Architecture
- 0203 Interior Design
- 0204 Landscape Architecture
- 0205 Urban Architecture
- 0206 City, Community, and Regional Planning
- 0299 Other, Specify

0300 AREA STUDIES
- 0301 Asian Studies, General
- 0302 East Asian Studies
- 0303 South Asian (India, etc.) Studies
- 0304 Southeast Asian Studies
- 0305 African Studies
- 0306 Islamic Studies
- 0307 Russian and Slavic Studies
- 0308 Latin American Studies
- 0309 Middle Eastern Studies
- 0310 European Studies, General
- 0311 Eastern European Studies
- 0312 West European Studies
- 0313 American Studies
- 0314 Pacific Area Studies
- 0399 Other, Specify

0400 BIOLOGICAL SCIENCES
- 0401 Biology, General
- 0402 Botany, General
- 0403 Bacteriology
- 0404 Plant Pathology
- 0405 Plant Pharmacology
- 0406 Plant Physiology
- 0407 Zoology, General
- 0408 Pathology, Human and Animal
- 0409 Pharmacology, Human and Animal
- 0410 Physiology, Human and Animal
- 0411 Microbiology
- 0412 Anatomy
- 0413 Histology
- 0414 Biochemistry
- 0415 Biophysics
- 0416 Molecular Biology
- 0417 Cell Biology (Cytology, Cell Physiology)
- 0418 Marine Biology
- 0419 Biometrics and Biostatistics
- 0420 Ecology
- 0421 Entomology
- 0422 Genetics
- 0423 Radiobiology
- 0424 Nutrition, Scientific (exclude Nutrition in Home Economics and Dietetics)
- 0425 Neurosciences
- 0426 Toxicology
- 0427 Embryology
- 0499 Other, Specify

0500 BUSINESS AND MANAGEMENT
- 0501 Business and Commerce, General
- 0502 Accounting
- 0503 Business Statistics
- 0504 Banking and Finance

The HEGIS Taxonomy

0505 Investments and Securities
0506 Business Management and Administration
0507 Operations Research
0508 Hotel and Restaurant Management
0509 Marketing and Purchasing
0510 Transportation and Public Utilities
0511 Real Estate
0512 Insurance
0513 International Business
0514 Secretarial Studies
0515 Personnel Management
0516 Labor and Industrial Relations
0517 Business Economics
0599 Other, Specify

0600 COMMUNICATIONS
0601 Communications, General
0602 Journalism (Printed Media)
0603 Radio/TV
0604 Advertising
0605 Communication Media (use of videotape, film, etc., oriented specifically toward radio/TV)
0699 Other, Specify

0700 COMPUTER and INFORMATION SCIENCES
0701 Computer and Information Sciences, General
0702 Information Sciences and Systems
0703 Data Processing
0704 Computer Programming
0705 Systems Analysis
0799 Other, Specify

0800 EDUCATION (Do not use; see 3100 and 3200.)

0900 ENGINEERING
0901 Engineering, General
0902 Aerospace, Aeronautical and Astronautical Engineering
0903 Agricultural Engineering
0904 Architectural Engineering
0905 Bioengineering and Biomedical Engineering
0906 Chemical Engineering (include Petroleum Refining)
0907 Petroleum Engineering (exclude Petroleum Refining)
0908 Civil, Construction, and Transportation Engineering
0909 Electrical, Electronics, and Communications Engineering
0910 Mechanical Engineering
0911 Geological Engineering
0912 Geophysical Engineering
0913 Industrial and Management Engineering
0914 Metallurgical Engineering
0915 Materials Engineering
0916 Ceramic Engineering
0917 Textile Engineering
0918 Mining and Mineral Engineering
0919 Engineering Physics
0920 Nuclear Engineering
0921 Engineering Mechanics
0922 Environmental and Sanitary Engineering
0923 Naval Architecture and Marine Engineering
0924 Ocean Engineering
0925 Engineering Technologies
0999 Other, Specify

1000 FINE and APPLIED ARTS
1001 Fine Arts, General
1002 Art (Painting, Drawing, Sculpture)
1003 Art History and Appreciation
1004 Music (Performing, Composition, Theory)
1005 Music (Liberal Arts Program)
1006 Music History and Appreciation (Musicology)
1007 Dramatic Arts
1008 Dance
1009 Applied Design (Ceramics, Weaving, Textile Design, Fashion Design, Jewelry, Metalsmithing, Interior Decoration, Commerical Art)

The HEGIS Taxonomy

1010 Cinematography
1011 Photography
1099 Other, Specify

1100 FOREIGN LANGUAGES
1101 Foreign Languages, General
1102 French
1103 German
1104 Italian
1105 Spanish
1106 Russian
1107 Chinese
1108 Japanese
1109 Latin
1110 Greek, classical
1111 Hebrew
1112 Arabic
1113 Indian (Asiatic)
1114 Scandinavian Languages
1115 Slavic Languages (other than Russian)
1116 African Languages (non-Semitic)
1199 Other, Specify

1200 HEALTH PROFESSIONS
1201 Health Professions, General
1202 Hospital and Health Care Administration
1203 Nursing
1204 Dentistry
1205 Dental Specialties
1206 Medicine
1207 Medical Specialties
1208 Occupational Therapy
1209 Optometry
1210 Osteopathic Medicine
1211 Pharmacy
1212 Physical Therapy
1213 Dental Hygiene
1214 Public Health
1215 Medical Record Librarianship
1216 Podiatry or Podiatric Medicine
1217 Biomedical Communication
1218 Veterinary Medicine
1219 Veterinary Medicine Specialties
1220 Speech Pathology and Audiology
1221 Chiropractic
1222 Clinical Social Work
1223 Medical Laboratory Technologies
1224 Dental Technologies
1225 Radiologic Technologies
1299 Other, Specify

1300 HOME ECONOMICS
1301 Home Economics, General
1302 Home Decoration and Home Equipment
1303 Clothing and Textiles
1304 Consumer Economics and Home Management
1305 Family Relations and Child Development
1306 Foods and Nutrition (include Dietetics)
1307 Institutional Management and Cafeteria Management
1399 Other, Specify

1400 LAW
1401 Law, General
1499 Other, Specify

1500 LETTERS
1501 English, General
1502 Literature, English
1503 Comparative Literature
1504 Classics
1505 Linguistics (include Phonetics, Semantics, and Philology)
1506 Speech, Debate, and Forensic Science (Rhetoric and Public Address)
1507 Creative Writing
1508 Teaching of English as a Foreign Language
1509 Philosophy
1510 Religious Studies (exclude Theological Professions)
1599 Other, Specify

1600 LIBRARY SCIENCE
1601 Library Science, General
1699 Other Specify

1700 MATHEMATICS
1701 Mathematics, General
1702 Statistics, Mathematical and Theoretical
1703 Applied Mathematics
1799 Other, Specify

The HEGIS Taxonomy

1800 MILITARY SCIENCES
- 1801 Military Science (Army)
- 1802 Naval Science (Navy, Marines)
- 1803 Aerospace Science (Air Force)
- 1899 Other, Specify

1900 PHYSICAL SCIENCES
- 1901 Physical Sciences, General
- 1902 Physics, General (exclude Biophysics)
- 1903 Molecular Physics
- 1904 Nuclear Physics
- 1905 Chemistry, General (exclude Biochemistry)
- 1906 Inorganic chemistry
- 1907 Organic chemistry
- 1908 Physical chemistry
- 1909 Analytical chemistry
- 1910 Pharmaceutical chemistry
- 1911 Astronomy
- 1912 Astrophysics
- 1913 Atmospheric Sciences and Meteorology
- 1914 Geology
- 1915 Geochemistry
- 1916 Geophysics and Seismology
- 1917 Earth Sciences, General
- 1918 Paleontology
- 1919 Oceanography
- 1920 Metallurgy
- 1999 Other, Specify

2000 PSYCHOLOGY
- 2001 Psychology, General
- 2002 Experimental Psychology (animal and human)
- 2003 Clinical Psychology
- 2004 Psychology for Counseling
- 2005 Social Psychology
- 2006 Psychometrics
- 2007 Statistics in Psychology
- 2008 Industrial Psychology
- 2009 Developmental Psychology
- 2010 Physiological Psychology
- 2099 Other, Specify

2100 PUBLIC AFFAIRS and SERVICES
- 2101 Community Services, General
- 2102 Public Administration
- 2103 Parks and Recreation Management
- 2104 Social Work and Helping Services (other than Clinical Social Work)
- 2105 Law Enforcement and Corrections
- 2106 International Public Service (other than Diplomatic Service)
- 2199 Other, Specify

2200 SOCIAL SCIENCES
- 2201 Social Sciences, General
- 2202 Anthropology
- 2203 Archeology
- 2204 Economics
- 2205 History
- 2206 Geography
- 2207 Political Science and Government
- 2208 Sociology
- 2209 Criminology
- 2210 International Relations
- 2211 Afro-American (Black Culture) Studies
- 2212 American Indian Cultural Studies
- 2213 Mexican-American Cultural Studies
- 2214 Urban Studies
- 2215 Demography
- 2299 Other, Specify

2300 THEOLOGY
- 2301 Theological Professions, General
- 2302 Religious Music
- 2303 Biblical Languages
- 2304 Religious Education
- 2399 Other, Specify

3100 EDUCATION (Physical Education)

3200 PHYSICAL EDUCATION (Other)

4900 INTERDISCIPLINARY STUDIES
- 4901 General Liberal Arts and Sciences
- 4902 Biological and Physical Sciences
- 4903 Humanities and Social Sciences
- 4904 Engineering and Other Disciplines
- 4905 Other, Specify

The HEGIS Taxonomy*

5000 BUSINESS and COMMERCE TECHNOLOGIES

- 5001 Business and Commerce Technologies, General
- 5002 Accounting Technologies
- 5003 Banking and Finance Technologies
- 5004 Marketing, Distribution, Purchasing, Business, and Industrial Management Technologies
- 5005 Secretarial Technologies (include Office Machines Training)
- 5006 Personal Service Technologies (Stewardess, Cosmetologist, etc.)
- 5007 Photography Technologies
- 5008 Communications and Broadcasting Technologies (Radio/TV, Newspapers)
- 5009 Printing and Lithography Technologies
- 5010 Hotel and Restaurant Management Technologies
- 5011 Transportation and Public Utility Technologies
- 5012 Applied Arts, Graphic Arts, and Fine Arts Technologies (include advertising design)
- 5099 Other, Specify

5100 DATA PROCESSING TECHNOLOGIES

- 5101 Data Processing Technologies, General
- 5102 Key Punch Operator and Other Input Preparation Technologies
- 5103 Computer Programmer Technologies
- 5104 Computer Operator and Peripheral Equipment Operation Technologies
- 5105 Data Processing Equipment Maintenance Technologies
- 5199 Other, Specify

5200 HEALTH SERVICES and PARAMEDICAL TECHNOLOGIES

- 5201 Health Services Assistant Technologies, General
- 5202 Dental Assistant Technologies
- 5203 Dental Hygiene Technologies
- 5204 Dental Laboratory Technologies
- 5205 Medical or Biological Laboratory Assistant Technologies
- 5206 Animal Laboratory Assistant Technologies
- 5207 Radiologic Technologies (X-Ray, etc.)
- 5208 Nursing, R.N. (less than 4-year program)
- 5209 Nursing, Practical (L.P.N. or L.V.N. - less than 4-year program)
- 5210 Occupational Therapy Technologies
- 5211 Surgical Technologies
- 5212 Optical Technologies (include Ocular Care, Ophthalmic, Optometric Technologies)
- 5213 Medical Record Technologies
- 5214 Medical Assistant and Medical Office Assistant Technologies
- 5215 Inhalation Therapy Technologies
- 5216 Psychiatric Technologies (include Mental Health Aide Programs)
- 5217 Electro Diagnostic Technologies (include E.K.G., E.E.G., etc.)
- 5218 Institutional Management Technologies (Rest Home, etc.)
- 5219 Physical Therapy Technologies
- 5299 Other, Specify

5300 MECHANICAL and ENGINEERING TECHNOLOGIES

- 5301 Mechanical and Engineering Technologies, General
- 5302 Aeronautical and Aviation Technologies
- 5303 Engineering Graphics (Tool and Machine Drafting and Design)
- 5304 Architectural Drafting Technologies
- 5305 Chemical Technologies (include Plastics)
- 5306 Automotive Technologies
- 5307 Diesel Technologies
- 5308 Welding Technologies
- 5309 Civil Technologies (Surveying, Photogrammetry, etc.)
- 5310 Electronics and Machine Technologies (TV, Appliance, Office Machine Repair, etc.)

* Reprinted from Huff and Chandler, The Taxonomy of Instructional Programs in Higher Education, Section 2: "Technological and Occupational Disciplines Leading to Associate Degrees and Other Awards Below the Baccalaureate."

5311 Electromechanical Technologies
5312 Industrial Technologies
5313 Textile Technologies
5314 Instrumentation Technologies
5315 Mechanical Technologies
5316 Nuclear Technologies
5317 Construction and Building Technologies (Carpentry, electrical work, Plumbing, Sheet Metal, Air Conditioning, Heating, etc.)
5399 Other, Specify

5400 NATURAL SCIENCE TECHNOLOGIES
5401 Natural Science Technologies, General
5402 Agriculture Technologies (include Horticulture)
5403 Forestry and Wildlife Technologies (include Fisheries)
5404 Food Services Technologies
5405 Home Economics Technologies
5406 Marine and Oceanographic Technologies
5407 Laboratory Technologies, General
5408 Sanitation and Public Health Inspection Technologies (Environmental Health Technologies)
5499 Other, Specify

5500 PUBLIC SERVICE RELATED TECHNOLOGIES
5501 Public Service Technologies, General
5502 Bible Study or Religion-Related Occupations
5503 Education Technologies (Teacher Aide and 2-year Teacher Training Programs)
5504 Library Assistant Technologies
5505 Police, Law Enforcement, Corrections Technologies
5506 Recreation and Social Work Related Technologies
5507 Fire Control Technology
5508 Public Administration and Management Technologies
5599 Other, Specify

The HEGIS Taxonomy

0000 Central Services

i.e., Services crossing support program categories

6600 Agriculture Extension Service

- 6610 Agriculture and Related Industries
- 6620 Social and Economic Development
- 6630 Quality of Living
- 6640 International Extension
- 6650 Unassigned
- •
- •
- •
- 6690

6700 Urban Extension Service

- 6710 Unassigned
- •
- •
- •
- 6790

6800 Reserved for Future Use

6900 Reserved for Future Use

7100 Student Development

- 7110 Student Activities
- 7120 Cultural Events
- 7130 Student Organizations
- 7140 Recreation
- 7150 Intramural Athletics
- 7160 (Reserved)
- 7170 (Reserved)
- 7180 (Reserved)
- 7190 Unassigned

7200 Intercollegiate Athletics

- 7201 Baseball
- 7202 Basketball
- 7203 Boxing
- 7204 Diving
- 7205 Fencing
- 7206 Football
- 7207 Golf
- 7208 Gymnastics
- 7209 Ice Hockey
- 7210 Lacrosse
- 7211 Rowing
- 7212 Rugby
- 7213 Skiing
- 7214 Soccer
- 7215 Swimming
- 7216 Tennis
- 7217 Track and Field
- 7218 Wrestling
- 7219 (Reserved)
- •
- •
- •
- 7249
- 7250 Unassigned
- •
- •
- •
- 7299

7300 Supporting Services

- 7310 Food Services
- 7320 Health Services
- 7330 Housing Services
- 7340 Retail Services and Concessions
- 7350 (Reserved)
- 7360 (Reserved)
- 7370 (Reserved)
- 7380 (Reserved)
- 7390 Unassigned

The HEGIS Taxonomy

7400 Special Student Services
- 7410 Veterans
- 7420 Foreign Students
- 7430 Disadvantaged
- 7440 Physically Handicapped
- 7450 (Reserved)
- 7460 (Reserved)
- 7470 (Reserved)
- 7480 (Reserved)
- 7490 Unassigned

8100 Central Operations
- 8110 Executive Direction
- 8120 Planning and Programming
- 8130 Legal Services
- 8140 Fiscal Control
- 8150 Investments
- 8160 Administrative Support
- 8165 Space Management
- 8170 Environmental Health and Safety
- 8175 Alumni Relations
- 8180 Community Relations
- 8185 Development
- 8190 Unassigned

8200 Functional Operations
- 8210 Fiscal Operations
- 8220 Student Admissions and Records
- 8230 Employee Personnel and Records
- 8240 Purchasing and Materials
- 8250 Communication and Services
- 8260 Transportation Services
- 8270 (Reserved)
- 8280 (Reserved)
- 8290 Unassigned

8300 Maintenance Operations
- 8310 Utility Plants
- 8320 Building Maintenance
- 8330 Grounds Maintenance
- 8340 Custodial Services
- 8350 (Reserved)
- 8360 (Reserved)
- 8370 (Reserved)
- 8380 (Reserved)
- 8390 Unassigned

8400 Plant Expansion and Modification (Noncapital)
- 8410 New Building Construction
- 8420 Facility Remodeling
- 8430 Landscaping and Grounds Modification
- 8440 (Reserved)
- 8450 (Reserved)
- 8460 (Reserved)
- 8470 (Reserved)
- 8480 (Reserved)
- 8490 Unassigned

9100 Other Instruction
- 9101 Unassigned
- •
- •
- •
- 9199

9200 Other Organized Research
- 9201 Unassigned
- •
- •
- •
- 9299

9300 Other Public Service
- 9301 Unassigned
- •
- •
- •
- 9399

9400 Other Academic Support
- 9401 Unassigned
- •
- •
- •
- 9499

9500 Other Student Services
- 9501 Unassigned
- •
- •
- •
- 9599

9600 Other Institutional Support
- 9601 Unassigned
- •
- •
- •
- 9699

9700 Institutional Operations
- 9701 Unassigned
- •
- •
- •
- 9799

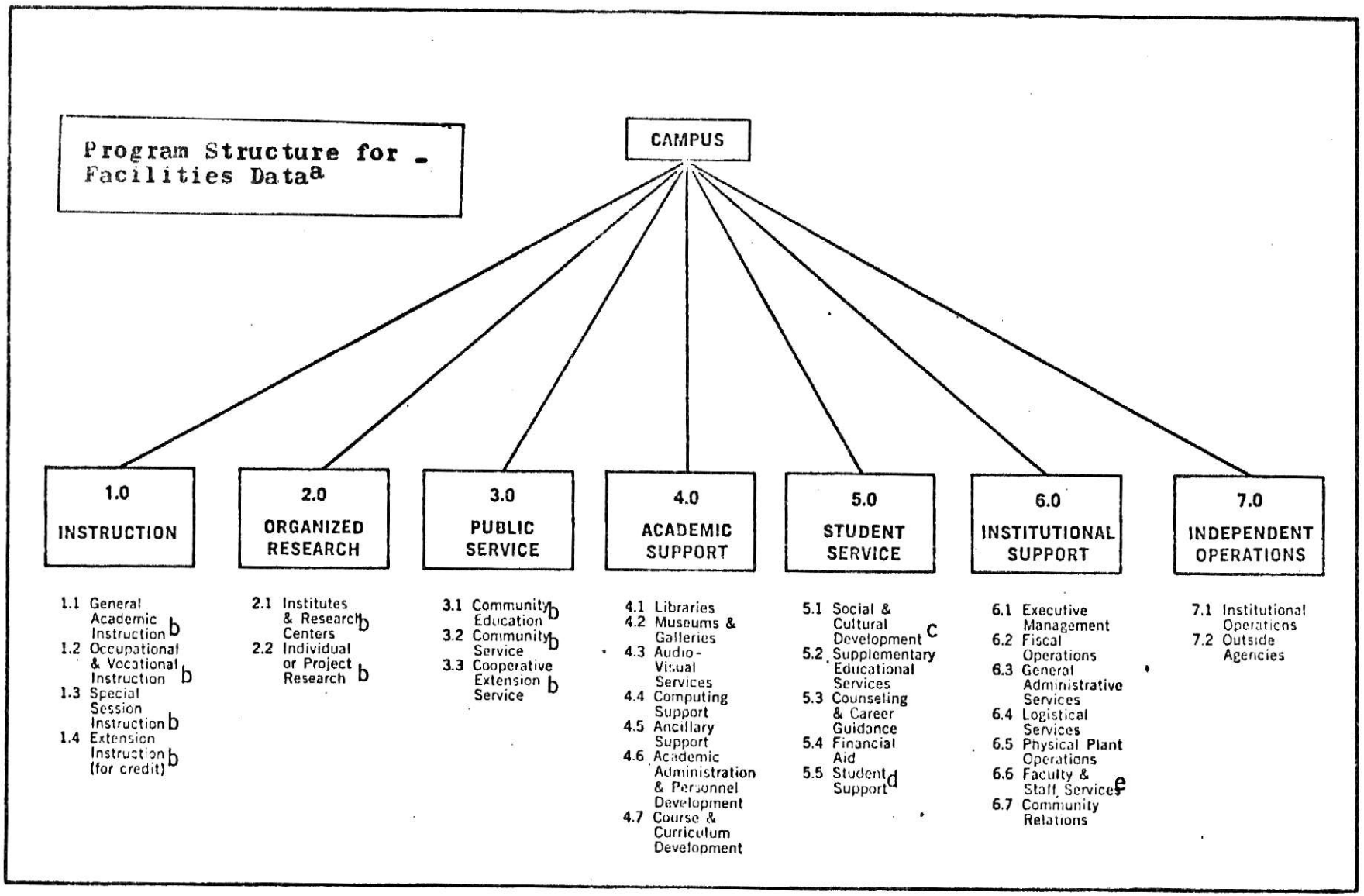
9800 Outside Agencies
- 9801 Unassigned
- •
- •
- •
- 9899

9900 Institutions Unique
- 9901 Unassigned
- •
- •
- •
- 9999

APPENDIX E

PROGRAM CLASSIFICATION STRUCTURE

Ref: National Center for Educational Statistics



[a]Depicts only the levels of the program structure recommended for facilities. For a complete explanation of the structure, see Warren W. Gulko, Program Classification Structure (Boulder: Western Interstate Commission for Higher Education, 1972).
[b]Further breakdown by HEGIS discipline categories (e.g., Physical Sciences) and disciplines (e.g., Physics). See Appendix 4.33 for a complete listing.
[c]Further breakdown to 5.1.72 for Intercollegiate Athletics.
[d]Further breakdown to 5.5.7330 for Student Housing Services.
[e]Further breakdown to 6.6.7330 for Faculty and Staff Housing Services.

APPENDIX F

ROOM TYPE CLASSIFICATION CODES

Ref: National Center for Educational Statistics

APPENDIX F

STANDARD ROOM USE CATEGORIES

100 CLASSROOM FACILITIES

- 110 Classroom
- 115 Classroom Service

200 LABORATORY FACILITIES

- 210 Class Laboratory
- 215 Class Laboratory Service

- 220 Special Class Laboratory
- 225 Special Class Laboratory Service

- 230 Individual Study Laboratory
- 235 Individual Study Laboratory Service

- 250 Non-Class Laboratory
- 255 Non-Class Laboratory Service

300 OFFICE FACILITIES

- 310 Office
- 315 Office Service

- 350 Conference Room (Office Related)
- 355 Conference Room Service (Office Related)

400 STUDY FACILITIES

- 410 Reading/Study Room

- 420 Stack

- 430 Open Stack Reading Room

- 440 Study Service

Appendix F

Standard Room Use Categories

500 SPECIAL USE FACILITIES

- 510 Armory
- 515 Armory Service

- 520 Athletic/Physical Education
- 523 Athletic Facilities Spectator Seating
- 525 Athletic/Physical Education Service

- 530 Audio/Visual, Radio, TV
- 535 Audio/Visual, Radio, TV Service

- 540 Clinic (Non-Medical)
- 545 Clinic Service (Non-Medical)

- 550 Demonstration
- 555 Demonstration Service

- 560 Field Service

- 570 Animal Quarters
- 575 Animal Quarters Service

600 GENERAL USE FACILITIES

- 610 Assembly
- 615 Assembly Service

- 620 Exhibition
- 625 Exhibition Service

- 630 Food
- 635 Food Service

- 650 Lounge
- 655 Lounge Service

- 660 Merchandising
- 665 Merchandising Service

- 670 Recreation
- 675 Recreation Service

- 680 Meeting Room
- 685 Meeting Room Service

Appendix F

Standard Room Use Categories

700 SUPPORTING FACILITIES

- 710 Data Processing/Computer
- 715 Data Processing/Computer Service
- 720 Shop
- 725 Shop Service
- 730 Storage
- 735 Storage Service
- 740 Vehicle Storage
- 745 Vehicle Storage Service
- 750 Central Food Stores
- 760 Central Laundry

800 HEALTH CARE FACILITIES

- 810 Patient Bedroom
- 820 Patient Bath
- 830 Nurse Station
- 840 Surgery
- 850 Treatment
- 860 Service Laboratory
- 870 Supplies
- 880 Public Waiting
- 890 Service

900 RESIDENTIAL FACILITIES

- 910 Individual Sleep/Study
  - 911 Sleep/Study without Toilet/Bath
  - 912 Toilet/Bath
  - 913 Sleep/Study with Toilet/Bath
  - 914 Sleep/Study Service

Appendix E

Standard Room Use Categories

920 Housekeeping

- 921 House
- 922 House Service
- 923 Apartment
- 924 Apartment Service

000 UNCLASSIFIED FACILITIES

- 050 Inactive Area
- 060 Alteration or Conversion Area
- 070 Unfinished Area

UNIVERSITY SPACE PLANNING:

PROJECTIONS FOR KANSAS STATE UNIVERSITY

by

K. CHANDRASHEKAR

B. E. (Mechanical Engineering)

University of Madras, India, 1970

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirement for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1973

The principal objective of this report is to develop a set of data processing programs to be used in the field of University Facilities Planning. This effort was planned to refine and restructure the work done by Claudill, Rowlett and Scott (CRS) - consultants to the Board of Regents, State of Kansas. Data processing programs were successfully developed and space projections were made for Kansas State University. The reports presented are more elaborate than those presented by CRS. However the author feels that this work is only an initial step and as such there is ample scope fur further improvements. No projection procedure has been developed for the Supporting, General Use, Medical Care and Residential Facilities. The input requirements have to be made less rigid. The entire work has to be thoroughly reviewed by the academic personnel involved with facilities planning.