A COMPUTER-BASED ADVISING AND INFORMATION SYSTEM
FOR COURSE SELECTION:
A PILOT STUDY

A Thesis
Presented to the
College of Graduate Studies
University of Saskatchewan
in partial fulfillment
of the requirements for the
Degree of Master of
BUSINESS ADMINISTRATION
by
ROSS M. HEWETT

September 1976
The author has agreed that the Library, University of Saskatchewan, may make this thesis freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this thesis for scholarly purposes may be granted by the professor or professors who supervised the thesis work recorded herein, or, in their absence, by the Head of the Department or the Dean of the College in which the thesis work was done. It is understood that due recognition will be given to the author of this thesis and to the University of Saskatchewan in any use of the material in this thesis. Copying or publication or any other use of the thesis for financial gain without approval by the University of Saskatchewan and the author's written permission is prohibited.

Requests for permission to copy or to make other use of material in this thesis in whole or in part should be addressed to:

Dean of the College of Commerce

University of Saskatchewan

Saskatoon, Canada.
ACKNOWLEDGEMENTS

The author expresses his appreciation to Mr. B.A. Holmlund who provided invaluable guidance and direction as thesis advisor. The author is also grateful to Dr. J.E. Cooke and Mr. G.D. Bicknell who were extremely helpful as members of the thesis committee. Sincere gratitude is also expressed to my wife, children and parents for their patience, understanding and encouragement.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 The Problem of Course Selection</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Needs of the Student When Selecting Classes</td>
<td>2</td>
</tr>
<tr>
<td>1.3 The Minimum Information System</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Existing Course Advising Systems</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Additional Information for Improved Decision Making</td>
<td>6</td>
</tr>
<tr>
<td>1.6 The Need for an Advanced System</td>
<td>7</td>
</tr>
<tr>
<td>1.7 The Ideal System</td>
<td>8</td>
</tr>
<tr>
<td>1.8 Objective of this Thesis</td>
<td>10</td>
</tr>
<tr>
<td>II. THE COUNSELLING FUNCTION</td>
<td>11</td>
</tr>
<tr>
<td>2.1 Counselling Defined</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Counselling at the Elementary and Secondary School Levels</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Counselling in Higher Levels of Education</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Computer-Aided Counselling Systems</td>
<td>15</td>
</tr>
<tr>
<td>2.5 Routine Course Selection</td>
<td>17</td>
</tr>
<tr>
<td>2.6 Two System Directions</td>
<td>18</td>
</tr>
<tr>
<td>2.6.1 Manual Course-Advising Systems</td>
<td>19</td>
</tr>
<tr>
<td>2.6.2 Computer-Based Course-Advising Systems</td>
<td>22</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>III. COMPUTER SYSTEM DESIGN ALTERNATIVES</strong></td>
<td>24</td>
</tr>
<tr>
<td>3.1 Two Computer-Based System Designs</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Batch Systems</td>
<td>25</td>
</tr>
<tr>
<td>3.2.1 Advantages of Batch Systems</td>
<td>27</td>
</tr>
<tr>
<td>3.2.2 Disadvantages of Batch Systems</td>
<td>27</td>
</tr>
<tr>
<td>3.3 On-line Systems</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1 Important On-line Considerations</td>
<td>33</td>
</tr>
<tr>
<td>3.3.2 Advantages of On-line Systems</td>
<td>36</td>
</tr>
<tr>
<td>3.3.3 Disadvantages of On-line Systems</td>
<td>36</td>
</tr>
<tr>
<td>3.4 Design Considerations for a Computer-Based Course-Advising System</td>
<td>37</td>
</tr>
<tr>
<td>3.4.1 Batch Processing</td>
<td>37</td>
</tr>
<tr>
<td>3.4.2 Remote Job Entry</td>
<td>38</td>
</tr>
<tr>
<td>3.4.3 Timesharing</td>
<td>39</td>
</tr>
<tr>
<td>3.4.4 Real-time Systems</td>
<td>39</td>
</tr>
<tr>
<td>3.5 The Selection of the Pilot System</td>
<td>40</td>
</tr>
<tr>
<td><strong>IV. THE PILOT STUDY</strong></td>
<td>41</td>
</tr>
<tr>
<td>4.1 Objectives of the Pilot Study</td>
<td>41</td>
</tr>
<tr>
<td>4.2 General System Description</td>
<td>41</td>
</tr>
<tr>
<td>4.3 The ADLIB Program</td>
<td>42</td>
</tr>
<tr>
<td>4.4 File Descriptions</td>
<td>42</td>
</tr>
<tr>
<td>4.5 The INFORMER Program</td>
<td>48</td>
</tr>
<tr>
<td>4.5.1 General Overview</td>
<td>48</td>
</tr>
<tr>
<td>4.5.2 INFORMER Logic Narrative</td>
<td>49</td>
</tr>
<tr>
<td>4.5.3 INFORMER Example</td>
<td>51</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th></th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6</td>
<td>The ADVISOR Program</td>
<td>62</td>
</tr>
<tr>
<td>4.6.1</td>
<td>General Overview</td>
<td>62</td>
</tr>
<tr>
<td>4.6.2</td>
<td>ADVISOR Logic Narrative</td>
<td>64</td>
</tr>
<tr>
<td>4.6.3</td>
<td>ADVISOR Example</td>
<td>70</td>
</tr>
<tr>
<td>4.6.3.1</td>
<td>Example 1--Non-major</td>
<td>70</td>
</tr>
<tr>
<td>4.6.3.2</td>
<td>Example 2--Major</td>
<td>84</td>
</tr>
<tr>
<td>4.7</td>
<td>Testing the Pilot System</td>
<td>95</td>
</tr>
<tr>
<td>4.7.1</td>
<td>The Test Plan</td>
<td>95</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Problems Encountered</td>
<td>98</td>
</tr>
<tr>
<td>V.</td>
<td>EVALUATION OF THE POLOT SYSTEM</td>
<td>100</td>
</tr>
<tr>
<td>5.1</td>
<td>Student Feedback</td>
<td>100</td>
</tr>
<tr>
<td>5.2</td>
<td>Analysis of ADVISOR's Suggestions</td>
<td>103</td>
</tr>
<tr>
<td>5.3</td>
<td>System Evaluation</td>
<td>106</td>
</tr>
<tr>
<td>5.4</td>
<td>Evaluation of the Batch Approach</td>
<td>109</td>
</tr>
<tr>
<td>5.5</td>
<td>Achieving the Ideal System</td>
<td>110</td>
</tr>
<tr>
<td>VI.</td>
<td>IMPLICATIONS OF THE STUDY</td>
<td>114</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions of the Pilot Study</td>
<td>114</td>
</tr>
<tr>
<td>6.2</td>
<td>Limitations of the Pilot Study</td>
<td>115</td>
</tr>
<tr>
<td>6.3</td>
<td>Implications of the Pilot Study</td>
<td>117</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY | 119 |
APPENDIX A - ADLIB Source Program Listing | 121 |
APPENDIX B - INFORMER Source Program Listing | 127 |
APPENDIX C - ADVISOR Source Program Listing | 132 |
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 MATRIX OF INTEREST WEIGHTS</td>
<td>68</td>
</tr>
<tr>
<td>5.1 RESPONSES TO FIRST QUESTIONNAIRE</td>
<td>101</td>
</tr>
<tr>
<td>5.2 COMPARISON OF ADVISOR'S SUGGESTIONS WITH STUDENTS' CHOICES</td>
<td>105</td>
</tr>
<tr>
<td>5.3 SUMMARY OF ADVISOR'S EFFECTIVENESS</td>
<td>107</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 The Problem of Course Selection

The success and reputation of a university's teaching program is largely determined by two major factors—what is taught and how it is taught. An excellent curriculum combined with a competent faculty will undoubtedly produce qualified and learned students.

The demands placed upon a curriculum today are enormous. A curriculum must contain classes that are general in nature so that students can be introduced to the field of study. As well, the curriculum must offer very detailed classes for the student who is interested in majoring in the field. The curriculum quite often has to allow for specialization in different facets of the total field. As a result, there is a very large number of classes offered to meet the demands of the various types of students.

In each discipline, the classes are usually arranged in a complex hierarchical structure, the completion of one class creating a link to another class of greater depth. These structures are not always clear to the inexperienced eye, and often get quite complex. Thus arises a problem that is inherent even in an excellent curriculum—course selection.
1.2 Needs of the Student When Selecting Classes

In order for a student to decide on a schedule of classes for the coming year, there is certain information that he must have at his disposal.

The student must know what classes he will be required to take, and at what stages in his university life they should be taken to meet his objectives in the shortest period of time.

The student also has a number of classes from which he must choose electives and classes towards his major. The student must know what classes are available, what are the prerequisites for each, and at what times they are offered. He would also like to know what the class is intended to cover, how it will be taught, and some idea of the reputation of the class.

The student must take the required classes at the times they are offered. If each required class has only one section, then the student's timetable becomes quite inflexible with the decisions being narrowed down to the selection of those electives which are offered at the open times.

However, if one or more of the required classes have two or more sections being offered at different times, the schedule becomes more flexible, and perhaps the decision making process becomes more difficult. The main basis for selection of electives will shift from time constraint to interest.

By consulting any information available about what happens in the class, the student decides whether or not he is interested in
in taking that class. When he encounters a class he is interested in, he checks to see if the times do not conflict with those of the required classes, and if he has satisfied the prerequisites for that class.

It is possible that two classes of interest to the student will have conflicting times or that there may be more classes of interest available than he can take. A value judgment based upon what he knows about the classes will determine which he will choose. It is, therefore, important that the information the student receives about the class content is complete and accurate.

1.3 The Minimum Information System

The very basic information system required in order for a student to select a class schedule is one which:

1) tells him the required classes he must take,
2) gives a description of electives available, and
3) gives the times at which the classes are offered.

Such a system will allow a student to construct a feasible timetable and meet the class requirements of the degree.

However, such a system will do nothing to ensure that the student receives the most out of his education. Many a student has enrolled in a class and learned very little because he wasn't interested in the subject matter of the class. It is possible that had the student received more information about the class, he would have selected another.
Therefore, a desirable system would be one that performs the functions of the above system and as well takes the interests of the student into account and arrives at a feasible timetable that will have maximum possible value to the student.

1.4 Existing Course Advising Systems

The present system at the University of Saskatchewan provides the basic information needed to construct a schedule of classes for the student. The College of Commerce, for example, has a section in the University calendar that includes:

- information about the College,
- a curriculum outline for each of the four years,
- the required classes for the degree,
- the required classes for each major,
- acceptable classes for each major,
- a paragraph description of each class, and
- the prerequisites for each class.

A timetable is supplied, giving the times at which each class is being offered. A worksheet is also provided for the student to record the classes he has already taken and thereby allows him to see any unsatisfied required classes as well as the number of electives that he will be required to take to complete his degree.

The student fills out his form with the required classes, chooses the electives he would like to take and attempts to arrive at a feasible schedule of classes for the next year. He then submits the
form to college personnel for approval. If he cannot arrive at a satisfactory schedule of classes, or wishes to obtain further advice, he is instructed to see one of the faculty or the Director of Undergraduate Programs in the College.

There have been problems in the past at pre-registration time as evidenced by the long line of students at the Director's Office, and at some of the faculty's offices as well.

Some of the problems have been caused by the student failing a required class, or the necessity to repeat some classes because of poor grades. For the student who has performed satisfactorily, the main problem is trying to decide what he should take for electives and/or classes for his major. The student goes through the lengthy sequence of obtaining information from the calendar, which may not be as accurate as it should be, then asking other students who may have taken the classes for their opinion, then talking to faculty members who are either teaching a particular class or may know something about it.

After spending considerable time in consulting these sources, the student may be faced with conflicting ideas, values and information. As a last resort, he turns to the Director of Undergraduate Programs and waits in line with numerous other students.

The net result of this series of events is often that a student may end up taking a class that he is not interested in, merely for the sake of getting the selection over with in as little time as possible.
1.5 Additional Information for Improved Decision Making

The basic information system described earlier is sufficient to allow the student to construct a feasible timetable. However, an increase in the amount of information available in certain areas will help improve the student's decision by allowing the student to evaluate the classes in light of his own interests.

For example, definite improvements would result if information was available in a structured manner such as:

a) description of class contents
b) objectives of the class
c) organization of the class
d) number and type of assignments and exams
e) performance of past students
f) times and instructors

With the exception of d) and e), portions of this information are presented in the calendar but not in a clear and simple manner. The breaking down of information in such a manner is important to allow evaluation and comparison of like classes.

Another way in which the decision process may be improved is facilitated by the use of an inverted timetable. This table would indicate to the student all classes offered at a certain time, on a certain day. Thus, if a student had certain time slots available for electives, he could easily determine what classes were offered at those times.
Probably the most pressing need for improvement in information lies in effectively and accurately expressing the contents of a class, in essence, what the student will be studying if he takes the class. A method of letting the student know exactly what is going to be covered in a class, and the emphasis that will be placed on each topic, must be found before an information system can be of maximum use to the student.

1.6 The Need for an Advanced System

Supposing it were possible to accumulate all of the information mentioned previously for those classes that a student is concerned with, do we merely throw it in his lap and let him spend hours trying to sort it out for himself?

Surely the answer is no. If this were to be done, the student would probably be bogged down with the volume of material and be left no better off than before.

A system has to be designed to help and guide the student toward making the best decisions for himself when selecting classes.

At the University of Wisconsin in 1968, Timmreck devised as part of his doctoral thesis "Advising by Computer," a course adviser for the College of Computer Science. The course adviser was a set of ALGOL programs that a student situated at a teletype could use. The adviser was broken down into two general parts, the interview and the suggesting of classes.

---

In the interview phase, the computer got information from the student about previous classes and grades and queried the student about his interests in the available topics.

In the actual advising phase, the program used a scheme of weights to rank those classes that student should be interested in from information obtained during the interview phase. The program generated a schedule of classes meeting appropriate criteria and asked the student whether he agreed. If not, the schedule would be adjusted until a satisfactory one was accepted by the student.

Timmreck's adviser is interesting in that it could handle routine problems such as required classes and prerequisites and also suggest those classes for electives that the student appeared to be most interested in.

The main drawback of Timmreck's adviser was the amount of time the system took to interview and finally advise the student—approximately one and a half hours for each student. Not only would it be expensive in terms of computer usage, but would also be quite an endurance test for the student. If such a system could function in fifteen or thirty minutes at the most, it would certainly be an improvement.

1.7 The Ideal System

The ideal university course advising system would be one which would:
1) be available at a number of locations on campus,
2) serve a student in any college on campus,
3) retrieve information about the student's past record (from a source other than the student),
4) give certain information upon demand about a class such as:
   - description of class contents
   - class objectives
   - class organization (seminar, lecture, labs)
   - number and type of assignments
   - number and type of examinations
   - class instructor
   - performance of past students
   - time offered
5) give a list of all classes offered at a certain time,
6) interview the student to find interests,
7) suggest a timetable of classes for the student taking into account
   - required classes
   - prerequisites
   - interests
   - time conflicts
8) be simple for student to use,
9) be easy for faculty to update information,
10) take less than one-half hour per student.
1.8 Objective of this Thesis

This thesis is a pilot study which has as its main objective the illustration of the more important functions of a course advising and information system comprising:

1) Informing the student about those classes required for the major;
2) Suggesting elective classes to the student that are most suited to his interests and needs;
3) Supplying detailed information about class content and structure upon demand.

Specifically, the study is limited to the design of a course advising and information system for the Department of Computational Science, an autonomous department of the University of Saskatchewan, with students enrolled from a number of other colleges.

The main reason for using the Department of Computational Science as the object of this study is the relatively small number of classes available. The study can be done on a smaller scale than if done for all of the College of Commerce, for example, and allows concentration on the most important functions of the system rather than requiring an enormous amount of data collection and associated problems.
CHAPTER II

THE COUNSELLING FUNCTION

2.1 Counselling Defined

The term "counselling" has been defined in a number of ways. One such definition of counselling is offered by Blocher:

Helping an individual become aware of himself and the ways in which he is reacting to the behavioral influences of his environment. It further helps him to establish some personal meaning for this behavior and to develop and clarify a set of goals and values for future behavior.¹

Counselling therefore requires two parties—the counsellor and the counsellee. The counsellor is usually someone with some amount of related training or experience that can be of help to the counsellee. The counsellee is a person who has a particular problem or a need for counselling and advice. According to Patterson,

A basic assumption of counselling services is that all individuals, from time to time, require specific personalized help of a professional nature in understanding themselves, the world, and themselves in relationship to the world, and in dealing with the immediate problems with which they are continually faced.²


2.2 Counselling at the Elementary and Secondary School Levels

Counselling at the elementary school level has essentially developed since the early 1960's. The fact that it is a recent development makes it also a rather controversial one. There are those who believe in the concept of the child development specialist or consultant. An opposing group feel that there is a logical extension of the secondary school counselling function into the elementary school. The important point that should be made is that the need for counselling of some sort at the elementary level has been recognized.

Counselling services offered at the secondary education level cover a broad spectrum. Following is a list of ten activities of the secondary school counsellor as adopted in 1964 by the American School Counsellor Association:

1. the planning and development of the guidance program and curriculum in relation to the needs of the pupils
2. the provision of counselling to help each pupil understand and accept himself as he is, develop personal decision-making competencies and resolve special problems
3. consultation and leadership in student appraisal activities
4. the collection and dissemination of information to pupils and parents regarding opportunities for further education and careers
5. the co-ordination of the use of special services beyond those provided by the counselling service through public relations with pupils, parents and outside agencies
6. the assistance in providing placement services for pupils for different levels of education and employment
7. the helping of parents by acting as a consultant to them regarding the growth and development of their children; providing them with information about their children and educational and occupational opportunities
8. the consultation with administrative and teaching staff
9. the conducting of local research into how well schools are meeting needs of local pupils
10. the carrying out of a program of public relations with the community.

The range of counselling services at the secondary level encompasses almost every possible aspect of the pupil's life while at school and attempts to prepare him for the future as well.

2.3 Counselling in Higher Levels of Education

Counselling in institutions of higher education can cover an equally broad spectrum as is indicated by Williamson:

The techniques of counselling individual students may be observed, in greatly modified form, in the individualized services for such problems as off-campus housing; granting loans and scholarships; handling discipline cases; assignment of rooms and selection of roommates in dormitories; advising on student...

---

activities and programs; helping students choose vocational objectives; selecting optional courses of study; learning to read at college rate and comprehension. Techniques, function, emphasis, and method are to be found in all types of service which deal with progress toward the goals selected by students, viewed as unique individuals.  

The rapid advancement of technology and related studies has led to the need for specialization of counselling services at the university level. "The role of counselling specialists will vary considerably from institution to institution, but will always be concerned with students who are experiencing difficulties in making adjustments to educational, vocational, or personal-social problems."  

Shaffer and Martinson have classified specialized counselling services into three main categories: educational, vocational and social-personal.  

Educational counselling entails the counselling of students by "assisting the student in developing realistic educational plans and in locating the various resources of the college or university that will aid him in attaining his objectives."  

Vocational counselling is aimed at helping the ill-prepared student who has not received the proper perspective of what preparation

---

6 Ibid.
7 Ibid., p. 44.
is required for a particular vocation, or alternatively, the student who, for some reason or other, is ill-informed about the occupation for which he has thought he was suited for.

Vocational counselling and the related area of appraisal are directed toward assisting the individual to gain a clear understanding of himself, his abilities and limitations; his sources of motivation; and his needs, attitudes and feelings.\(^8\)

Personal-social counselling is the term most frequently used to describe problems associated with a concern about an individual student's interactions with others or a feeling of insecurity or inadequacy on his part. "Although any of the techniques used in other types of counselling may be employed, the primary technique is that of the interview in which the student is encouraged to gain greater understanding of himself and his relationship to the problem situation."\(^9\)

2.4 Computer-Aided Counselling Systems

The previous sections of this chapter have attempted to define and describe the counselling function. The intent of this section is to give a brief introduction via some examples of the extent to which computers have been used to aid in the counselling process.

In a growing number of areas, computers are being used as an aid in an otherwise human process. This trend is somewhat of a middle road approach resulting from situations where computers were not able

\(^8\)Ibid., p. 45.

\(^9\)Ibid.
to perform the function completely by themselves, but when combined with human intelligence and perception, resulted in an overall improvement over either a totally manual or computerized system.10

Such is the case with computer-aided counselling systems. The human interaction and subjectivity required in the counselling process is such that it has not been satisfactorily duplicated by a computer system alone. Systems have been developed, however, where the computer plays an important role as a tool for the human counsellor.

A system referred to as TGISS (Total Guidance Information Support System) was developed at Oklahoma State University in 1972 for use by counsellors at the secondary school level. The system displays upon request, through CRT's to a counsellor, personal and academic information about a student who is being counselled. As well, the system was available to the student to retrieve occupational information.11

A similar system was developed for university level, and specifically for colleges of education by Rebhun, Bonar and Dick.12

Some computer-assisted counselling systems have also been integrated with the registration process. Johnson County Community College

in Shawnee Mission, Kansas, has a system developed in conjunction with a local computer service bureau that uses CRT's for the counsellor to retrieve student information for the counselling process, but in addition, the counsellor can register the student in various classes through the CRT. The system will also do some checking to ensure that the prerequisites, for example, have been satisfied for the classes the student is registered in.  

Michigan University in Ann Arbor, developed in 1973 a system they referred to as CRISP (Computerized Registration Involving Student Participation). It was designed primarily as a registration system with the purpose of reducing the registration process. The system consists of terminal operators who key in classes requested by the students and the system reserves those classes if available. Thus, the system takes over after the student has decided on what classes he wants rather than actually helping a counsellor or advising the student directly.  

2.5 Routine Course Selection

The counselling function, as has been pointed out previously in this chapter, is primarily concerned with helping those students with
certain educational, vocational and personal-social problems. Often the result of such counselling is that the student has identified what his interests are, and what college and major area of studies are best suited to the fulfillment of his objectives. Occasionally, if required, the counsellors will help the student determine which classes he should take, or alternatively, will refer the student to faculty in the various departments for advice on course selection.

There are those students that do not feel they require any counselling and have already decided the college and major that they want to achieve. The problems that they encounter are related to understanding the prerequisite structure and required classes, etc., as eluded to in the Introduction.

The intent of a course-advising system as being studied by this thesis is not to replace the human counsellor, but rather to eliminate the confusion and simply the problem of routine course selection. The students who are having problems in deciding the college or major should still seek the advice of the counsellor.

2.6 Two System Directions

The need for an improved course-advising and information system was established in the Introduction. Ten requirements of such a system were outlined in 1.7 The Ideal System, but the type of system that should be used has not been established.

There are two directions or types of systems in general--manual and computer-based. The manual system is comprised of human beings who
do all of the work and are sometimes aided by mechanical devices, as compared to a computer-based system where the computer does the majority of the work and may be occasionally aided by human beings.

2.6.1 Manual Course-Advising Systems

The main component in a manual course-advising system is the counsellor. The counsellor may have at his disposal such mechanical information-retrieval devices as microfilm or microfiche as well as conventional hard copy paper reports. Following is an example of how the manual course-advising and information system might operate, keeping in mind the previously developed requirements of the ideal system.

There would be a number of counsellors on the campus. Each counsellor would probably be assigned to a certain college or department and there may be a number of counsellors in each area depending upon enrollment. The students would consult an appropriate counsellor during the registration process. The counsellor would first of all obtain the student's name and student number and then look up the student's record on either a paper or microfiche file. After determining what major the student is striving for, the counsellor would then present that student with a number of well selected interest topics for that major, and ask him to rate his interest in each topic. Upon completion of this task by the student, the counsellor would match these interest ratings against the relative coverage by the offered classes. The matching would result in an interest weighting for each class. The next step in the manual system would be the identification by the counsellor of
the classes that remain to be taken to satisfy degree and major requirements. These may be the actual required classes themselves, or their prerequisites if not already taken. The next step is the satisfying of elective requirements, based on the student's interests as determined by the interest-topic-matching process described earlier.

The final process would be the construction of the student's timetable. The timetable would then be presented to the student for consideration. The reasons why each class is on the timetable would be explained to the student. Time conflicts and possible solutions would have been discussed with the student during the construction of the timetable. The student might have to make choices between classes and to do so may require additional information about each class. The counsellor would have that information in either printed catalogue form or on microfiche, and he and the student would discuss it together with the counsellor perhaps being able to give some advice based on his subjective analysis of the situation. The timetable would finally be approved by the student and the process would be completed by the student leaving the counsellor and going to register in the individual classes. The next student would then meet with the counsellor and the process would be repeated.

In comparing this example of a possible manual course-advising system with the ten requirements of the ideal system, one can see that all of the requirements could be met, with the possible exception of
the last, i.e. less than one-half hour per student. The following
gives an analysis of the duration of the process using crude estimates
of times of the various components, as developed by the author of this
thesis.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>1. locate student record</td>
<td>2</td>
</tr>
<tr>
<td>2. student rating of interest topics</td>
<td>5</td>
</tr>
<tr>
<td>3. matching and weighting of classes for interest</td>
<td>10</td>
</tr>
<tr>
<td>4. identification of remaining required classes</td>
<td>2</td>
</tr>
<tr>
<td>5. selecting electives</td>
<td>5</td>
</tr>
<tr>
<td>6. providing class information</td>
<td>5</td>
</tr>
<tr>
<td>7. determine timetable</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>39</td>
</tr>
</tbody>
</table>

As can be seen from the above estimates, the range is from 39
to 85 minutes or an average counselling time per student of approxi-
mately one hour. It should be made clear that these are estimates only,
and are presented merely to indicate that the manual system might be
quite time consuming.

It is apparent that the manual system could actually provide
an improved course-advising and information system. However, it is
also possible that such system would require a considerable number of
qualified people. As a result, the cost of such a service could conceivably be very high. For example, if one counsellor was added for each department at the university, the cost for the College of Arts and Science alone would be in excess of $100,000 per year.

2.6.2 Computer-Based Course-Advising Systems

The computer-based course-advising system has as its main component a computer system made up of various pieces of hardware and software.

Computer applications typically get their existence from duplicating routine manual operations at lesser cost and with perhaps better results than the manual system. The functions that were described earlier in the manual course-advising system and that were performed by the counsellor and his mechanical devices lend themselves quite readily to the process of computerization. To review, the counsellor's first task was to retrieve the student's academic record. Student records could be placed in a large computer file and accessed by the course-advising system's programs by using the student number and name provided to the system by the student. As well, the student could identify to the computer system which college or department he is interested in, what his major is and also provide the system with his ratings on interest topics in a variety of ways. The computer system's programs would be able to compare those ratings with coverage ratings previously provided by the faculty and automatically perform the
interest weighting function. The computer system could also be
programmed to be sensitive and cognizant of the complex prerequisite
class structure.

The computer-based system could also be programmed to set up a
timetable for the student in much the same manner as would the human
counsellor, but with a high probability of doing it much faster than
would be humanly possible. Information retrieval on class description
could be done much more quickly through the computer than using printed
material.

In essence, the computer-based system should be able to do every-
thing the manual system could with one exception—the computer-based
system cannot use subjectivity to relate to the student. This, however,
may not always be a disadvantage in that human judgment or biases can
sometimes be in error and not in the best interests of the student.
CHAPTER III

COMPUTER SYSTEM DESIGN ALTERNATIVES

3.1 Two Computer-Based System Designs

The need for the advising system to be computer-based has been established in the previous chapter. The intent of this section is to describe the design alternatives that are available.

Batch processing is a term that has been used to describe the traditional method of data processing. It has been defined as a system in which "a large number of transactions are gathered together into a batch. At regular intervals the programs and files which are needed are loaded and the transactions processed."¹

On-line processing, the second alternate system design, is a significantly different approach. The term "on-line" has been used to indicate that there is a direct link between the user and the system. An on-line system has been defined as "one in which the input data enter the computer directly from their point of origin and/or output data are transmitted directly to where they are used."²

3.2 Batch Systems

Batch systems, as previously defined, are the traditional data processing approach. Like transactions are gathered into a batch and submitted to the computer for processing. The time delay between when a transaction is created until it has been processed can be any length with a minimum of a few minutes to a long delay.

In the early stages, batch processing was mainly sequential in nature—for example, the transactions would have to be pre-sorted into the same sequence as the master file that was to be updated. This sequential processing was mainly a requirement imposed by sequential files resident on tapes or punched cards.

With the advent of direct access storage devices such as magnetic disks, sequential processing was combined with what has been termed in-line processing. "The term in-line comes from the fact that one can save the transactions and then feed them into the program in the sequence in which they were created."\(^3\) With this new capability, some of the time-consuming sorting activities were eliminated from batch processing.

The components of a batch processing system are as follows:

1. Input data preparation
2. Hardware

3. Software

4. Output preparation

The input data preparation component performs the functions of collecting the raw input data and preparing that data in a form suitable for processing by the system. In some cases, the input for a subsystem may be the output from another system, in which case the need for this component for the subsystem is negated. However, for the batch system in total, this component must exist. The make-up of this input preparation component is a combination of people and machines which is usually referred to as the data entry group. The data entry group utilizes such devices as key punch, key to tape, key to storage, and key to disk to take the raw data from source documents and transform it into acceptable input for the computer.

The hardware component of the batch processing system refers to the basic computer components such as the I/O devices, CPU, and auxiliary and main storage devices.

The software component consists of two main parts:

1. operating system, and

2. application programs.

The operating system is responsible for controlling the workload of the system and resource allocation. The application programs consist of instructions that perform the actual functions of processing the input data, updating files and producing the required output.
The output preparation component of the batch processing system is similar to the input data preparation component in that it typically is made up of a man-machine combination. The function of this component is to decolate, burst, sort and distribute the reports that are produced by the system as well as to maintain libraries of output in other formats such as tapes.

3.2.1 Advantages of Batch Systems

1. Efficiency

The entire concept of batch processing rallies around the fact that it is more efficient to process a like number of transactions together than it is to process them interspersed with other transactions, thus minimizing the overhead from loading programs a number of times.

2. Workload balancing

Batch processing allows the workload of the system to be balanced by scheduling certain applications to be handled in non-peak hours. This can result in a better utilization of computer resources.

3.2.2 Disadvantages of Batch Systems

1. Timeliness

The batch system concept often entails a delay of hours between the time the work is submitted until the results
are received. In some cases, the delay makes the information useless.

2. Cost of I/O preparation

The cost of I/O preparation may be reduced or eliminated if the input could be captured at the source and the output automatically returned to the source.

3.3 On-line Systems

The discussion of batch processing systems revealed one of its disadvantages as being timeliness of information. This factor has been referred to as the system response time. Response time is defined by Yourdon as "the time required to process an input and deliver the desired output." The response time question is one of the largest reasons for the existence of on-line systems.

There are a number of versions of on-line systems. Remote job entry (RJE) is a good example where improvements have been made to the batch processing system by placing an I/O terminal in the hands of the user and allowing him to submit a job directly to the computer and receive the output back at the terminal. However, unless there is a partition or computing resources dedicated to that terminal, his request will have to compete with other jobs that have been previously

---

submitted for resources and the response time may therefore be lengthy. The I/O preparation functions have been transferred to the user, which will allow him more control over delays in this area.

Timesharing is another version of an on-line system. In timesharing systems, users are given "time slices" or small portions of computer execution time in which portions of their job are executed while the other users wait. In this fashion, each user feels he has control of the computer at the other end. In order to handle the terminals and maintain security as well as storage of partially completed tasks, a supervisor program is required to oversee the timesharing operating. Since the supervisor itself uses part of each time slice, the timesharing system actually requires more work to process the jobs in comparison to batch processing, but because of the elimination of keypunching, etc., "the average user sees a considerably shorter apparent response time in a timesharing system than in a centralized batch system."5

Another version of an on-line system is the real-time system. One example of a real-time system is a process control system, where data is collected through sensors, processed and adjustments are made to the manufacturing or similar process. Real-time systems have been defined as one "which controls an environment by receiving data, processing them and returning the results sufficiently quickly to

affect the functioning of the environment at that time.\textsuperscript{6} Real

*time systems have come to mean more than just process control systems. Information retrieval systems are a good example of the application of the concepts of real-time systems outside of process control. The underlying identifying characteristic of this type of real-time system is the absence of the submission of a program as input. Instead, a code or key is submitted to an awaiting program that will provide a "cue" to retrieve some information or start up some interactive process.

There are two basic types of this kind of real-time system—

inquiry and interactive. The inquiry system is simple in that once the reply is made, the task is completed. However, with interactive systems, a much more complex set of conditions is imposed upon the system because of the fact that the system may have to remember previous inputs from the user to be used later in the process.

The on-line real-time system can be divided into three fundamental or basic components: the hardware, the programming system, and the users. As well, there are interfaces between the hardware and the programs as well as between the programs and the users. Each of the three basic components can be further divided into sub-components.

The sub-components of hardware are: the terminal, communications linkage, the central processing unit and main memory, and direct access devices.

The terminal is the piece of hardware that the user sees and is his means of communication with the system. The terminal can take many forms, from the common slow speed teletype to more elaborate CRT devices equipped with light pens. The type of terminal is highly dependent upon the particular needs of the user.

The communication linkage can take a variety of forms and is responsible for conveying the information between terminals and the central processing unit. This device is often a small mini-computer and handles all of the error checking, polling and other terminal control functions and passes the data to and from the central processor.

The central processing unit is, of course, the "brains" of the system which performs all arithmetic and logical operations as instructed by the system programs. The main memory is used by the CPU for storing programs and data currently being processed. The amount of main memory has a definite effect on processing times.

The direct access storage devices (DASD's) provide auxiliary store for program segments and data files. Sequential access devices such as tape drives are occasionally used for logging transactions for back-up purposes. DASD's are used because of their fast access times as well as random access capabilities, qualities necessary for on-line interactive systems.

The programming system component can be broken down into three main sub-components: the supervisory programs, application programs and support programs.
The supervisory programs co-ordinate and schedule the work of the application programs and carry out service functions for them. The supervisory programs handle input and output operations, and exercise control over the total on-line system allocating resources and responding to various interrupts for service. As well, it is the supervisory programs function to look after and deal with error and emergency situations.

The application programs are the programs that carry out the work or perform the functions as required by the user. They correspond to application programs of the batch processing system and are unique to each individual system or application in direct contrast with the supervisory programs which are generalized to handle a number of different application programs. The application programs do not get involved in I/O operations except to generate macro "get" and "put" commands to the supervisory programs at the required times.

The support programs are mainly concerned with maintenance functions such as system generation, diagnostics, and terminal simulators for testing.

Control can pass from the application programs to the supervisory programs in one of two ways: interrupts and macro calls.

Interrupts can typically be caused by the following:

1. A signal that an input/output operation may begin (in response to a previous request for that I/O operation).

---

2. A signal that an I/O or file operation is complete.

3. A signal that an error or abnormal condition has been detected.

4. A signal that indicates a special condition such as a timer completing its cycle.

The interrupts can also be assigned priorities so that higher level interrupts can interrupt lower levels, with the end result of maximizing the use of I/O devices.

Macro calls are instructions that have been coded by the application programmer to cause deliberate linkages between the supervisor and the application program. The macro instructions pass control back to the supervisor for situations where processing cannot continue for some reason, such as the completion of the task, or the need to wait for the retrieval of a file record.

3.3.1 Important On-line Considerations

1. File organization

On-line systems, as mentioned previously, will make extensive use of direct access storage devices and rather limited use of sequential devices. DASD's may be organized in one of three ways: sequentially, indexed sequentially, and randomly (or directly). These three methods are commonly used in batch processing systems. However, in on-line systems, where response time is of great importance, the method providing the fastest access is required. The direct access
approach provides the fastest access method, but usually results in wasted storage space. Indexed sequential is slightly slower, but makes more efficient use of disk space and is more often chosen over direct organization.

As of late, there has been a swing to the data base approach. "The totality of data that is stored in a system is frequently referred to as the data base of the system." The function or method that controls access to this data base whether for entering new data or extracting needed data, is referred to as data management.

The primary reason for the popularity of the data base concept lies in the advantage of data independence which has been defined as "the divorcing of the description or definition of data from any individual program, thus allowing multiple users and programs to access the same data files without undue difficulty." 9

The previously mentioned file organization methods (index sequential, direct) relate to the organization of one file and do not contain the facility for accessing or linkage to data in other files, as is necessary with the data base concept. A number of methods of connecting items of data are available, such as linked lists, threaded lists, inverted files, tree-structured indexing and ring structures. A number of data base management systems have been commercially

---

9 George Schuessel, "When Not to Use a Data Base," Datamation, Nov. 1975, p. 82.
developed employing these concepts and have been developed to be used primarily in conjunction with on-line information retrieval systems.

2. The man-machine interface

The man-machine interface encompasses the entire method in which the user interacts with the system and includes his expectations of what he feels the system can do for him. The main piece of machinery that the user interacts with in an on-line system is the terminal. If the terminal is not suited for the application, then the success of the system is in jeopardy. Also, if the dialogue with the user is not appropriate or is inefficient, the overall effectiveness of the on-line system can be degraded.

Today there are many, many types of terminals available. As well, there are just as many, if not more, types of dialogue that can be designed into a system. Martin lists 18 different types of dialogue, each with its own advantages and disadvantages. 10

The important point that should be made is that the type of dialogue and terminal that are to be used in an on-line system should be given considerable thought, and will vary depending upon the application for which the system is intended.

3.3.2 Advantages of On-line Systems

1. Timeliness

On-line systems have the capability of providing virtually instantaneous response times, which for some applications, are very valuable and perhaps even necessary.

2. Elimination of intermediate actions

On-line systems, because of their nature, eliminate the need for the I/O preparation functions of the batch system and its associated costs.

3. New information tools

Combined with database management systems, graphic displays, etc., new frontiers such as interactive problem solving, for example, are opened that would not be possible in the batch processing environment.

3.3.3 Disadvantages of On-line Systems

1. Cost

The addition of elaborate hardware and software over and above the batch processing system may result in increased investment in the computer resource.

2. Error recovery

Error recovery in batch processing is quite straightforward and consists of merely using back-up files and rerunning the jobs.
Re-running all jobs is practically impossible in an on-line environment and, therefore, sophisticated error recovery routines and/or duplicate hardware must be acquired.

3.4 Design Considerations for a Computer-Based Course-Advising System

The previous section gave a brief overview of two system design alternatives—batch processing and on-line processing. It further defined on-line systems as being either RJE, timesharing or real-time systems. The intent of this section is to relate these alternatives to the problem at hand, the design of a computer-based course-advising system.

In order to evaluate the system design alternatives, it may be beneficial to attempt to visualize how a course-advising and information system would operate in each case.

3.4.1 Batch Processing

In a batch processing system, a number of student requests would be collected and processed together at various time intervals. The output would be in printed form and would be identified for each student.

The student's request must, of course, be in some type of machine readable form to be processed by the system. This function must be performed by either the student or the institution. The most practical alternative would be to have the student fill out a request form in writing and submit that form to a central location.
for preparation as input. The student could also pick up his output at the central location as well.

In analyzing the batch approach, two main drawbacks can be identified. The first problem is the delay between the initiation of the request, and the supplying of the output. At this point, it is unknown whether there is a need for a quick response time or not. It is possible that a few hours' delay could lessen the usefulness of the system to the student. The second problem is that of the data entry function. The batch system requires that both people and equipment be available to collect and process input and to distribute the output.

3.4.2 Remote Job Entry

A course-advising and information system operating under the RJE concept would entail RJE terminals in a number of locations on the campus—in each college office for example. The terminal would consist of an input device, such as a card reader, and an output device, probably a line printer. Using the card reader terminal for example, the students' requests would be keypunched, transmitted to the host computer, processed and the output printed back on the line printer. The response time could be in terms of seconds, depending upon the availability of computer resources.

The RJE system, therefore, may overcome one disadvantage of the batch system—poor response time. However, the data entry problem still remains.
3.4.3 Timesharing

In the timesharing system a student would sit down at a terminal and log on to the system. (The log-on procedure could be explained in simple terms to aid wary students.) The system would accept the student's requests through the terminal, process them and print the output back at the student's terminal in hard copy form for the student to remove and study at his leisure.

The timesharing system has therefore eliminated the disadvantage of the RJE system by allowing the student to perform the data entry function himself. This is possible because of "lead through" techniques that would guide the student through the task of entering his request and allow him to correct errors as they occur.

However, the timesharing system contains one serious drawback--the investment in terminal equipment. The number of terminals required to satisfy peak demand periods may require a substantial investment such that this system could become uneconomical.

3.4.4 Real-time Systems

The real-time system would also have the student enter his own data through terminals. Although certain economies can be attained as compared to timesharing, the number of terminals required could still be substantially high enough to offset any advantages over the RJE or batch systems.
In summary, each system type has its disadvantages: the batch system has a long response time and requires a data entry function; the RJE system solves the response problem but still requires the data entry function; the timesharing and real-time systems require a heavy investment in large volumes of terminal equipment.

As a result, it appears that the choice of the system design alternative is mainly an economic one and may be influenced by other computing applications at the campus.

3.5 The Selection of the Pilot System

The previous section discussed the merits of the various system design alternatives but could not draw a conclusion as to which alternative was the best for a course-advising and information system.

As was discussed in the Introduction, this thesis is a pilot study which has as its main objective the illustration of the more important features of a course-advising and information system. It has become evident that any one of the four system alternatives would allow this objective to be achieved.

On this basis it was decided to adopt the batch system approach for the pilot study, mainly because of the lower development costs of the batch approach as compared to the other alternatives.
CHAPTER IV

THE PILOT STUDY

4.1 Objective of the Pilot Study

The objective of this thesis was stated in the Introduction as the illustration of the more important functions of a course-advising and information system. In order to meet this objective, a pilot study was conducted. The pilot study consisted of two main parts:

1) the design and development of a pilot course-advising and information system

2) the testing of the pilot system using real students

As was mentioned in the Introduction, the study was limited to the design of a course-advising and information system for the Department of Computational Science.

The development phase of the pilot study took place over the summer of 1971. The testing of the system was performed in September of the same year, using a sample of students enrolled in the College of Arts and Science and the College of Commerce.

4.2 General System Description

The pilot system consists of three FORTRAN programs. ADLIB is a maintenance program that accepts input data via punched cards and builds a total of twelve files on disk to be used as input by the other two programs.
INFORMER is an information retrieval program that accesses eight of the files in the system and produces printed output in response to student requests for information submitted on punched cards.

ADVISOR is the program that handles the advising function of the system. It accesses seven files, three of which are shared with INFORMER. The student data is submitted via punched cards and the output is in printed form.

4.3 The A DLIB Program

ADLIB is designed to construct and/or update the system files. It has been written in FORTRAN and requires that the input be in certain fixed formats and error-free.

ADLIB will allow selected files to be updated individually, but is not flexible enough to allow selective updating of individual records of a file. This is not a serious limitation due to the fact that it is a pilot study only, and convenience of file updating is not of prime importance. The ADLIB program creates a printed report of the contents of each file that has been created.

A source program listing of ADLIB is presented in Appendix A.

4.4 File Descriptions

ADLIB creates twelve separate files in total, each residing on disk. A brief description of each file is presented below.
The following four files are used exclusively by INFORMER using the direct organization approach. They key to a class's record is equal to a predetermined sequence number. Each record contains a maximum of 1000 characters of text.

1. class description file
   - this file contains a general description of each class.
   The performance of the students in the past term is also included in this file when available.

2. class objectives file
   - this file contains a description of the objectives of each class being offered.

3. major topics file
   - this file presents a list of the major topics or area of study to be covered by a class.

4. class organization file
   - this file describes how the class is organized, whether there are laboratory periods, the number of examinations that will be expected, etc.

5. inverted time file
   - this file is accessed only by INFORMER and is used to provide information on all classes offered at a certain time. The file is organized in a direct fashion with
one record for each possible starting time for each day and for both A and B terms. The location of the desired record is calculated by an algorithm and the record is then read directly from the disk. ADLIB requires that a card be present for each starting time, and each class that is offered at that time has its sequence number punched on the card.

6. class list file

- this file is used by both INFORMER and ADVISOR. The file is organized sequentially, one record per class. The record consists of four fields: an eight character class name (such as CMPT 377), a two digit synonym code, a one digit class type code, and a sixty character class title (such as DATA PROCESSING). The synonym code (when non-zero) indicates a class is identical to another class (the parent class) and is set equal to the parent class's sequence number. The parent class's synonym class remains at zero. The class type code indicates whether the class is a full class, a half class, or a class offered by another department. The sequence number of a class is defined as being equal to the relative record location in the class list file. Therefore, if CMPT 377 is the fifth record in the class list file, then that class's sequence number is equal to 5. The sequence number is the basis
for referring to classes within the programs and the files. The actual class name (CMPT 377) is only used for input and report purposes.

7. time index file
- this file is also used by both INFORMER and ADVISOR. This file serves as an index to locations of class records in the time file. The file contains one record for each class. The record contains the location of the corresponding record in the time file.

8. time file
- this file is accessed by both INFORMER and ADVISOR after retrieving the desired record location from the time index file. The desired record is read directly from disk. There is one record for each section offered. Each record contains the term, the section number, the days, times, room, and instructor.

9. prerequisite file
- this file is accessed by ADVISOR only. The purpose of the prerequisite file is to identify which classes have prerequisites, what classes are involved and what, if any, combinations or choices are available. The prerequisite file makes use of a logical condition code which indicates the type of prerequisite structure. The codes that fitted
all existing conditions at the time the system was
developed are as follows:

0 no prerequisites
1 P1 (only one prerequisite)
2 P1 or P2
3 P1 and P2
4 P1 and P2 and P3 and P4
5 P1 or P2 and P3 and P4
6 P1 and P2 and P3
7 P1 or P2 and P3

The file consists of one record per class, each record
consisting of five fields—the logical condition code
and the sequence-numbers of the four prerequisite classes.
The file is in the same sequence as the class list file.

10. required classes file

—this file is used by ADVISOR only. In the Department of
Computational Science, and most other departments and
colleges, there are classes that must be taken if a student
is to major in a certain area. These required classes are
of two forms. The first are those classes the student must
take. The second occurs when a student must choose a mini-
mum number of classes from a given set of classes. Required
classes may also vary according to the student's college.
The required classes file contains three records for each applicable college that offers a major in Computer Science. The first record contains a college identifier and the total number of required classes. The second record indicates the "required options" which refers to the case where a student must choose a minimum number of classes from a given set. The minimum number as well as the sequence numbers of the available choices are contained in the record. If the options do not exist, the fields have a value of zero. The third record contains the sequence numbers of the required classes.

11. interest topic coverage file
- this file is used only by ADVISOR. The file contains one record for each class and is in the same sequence as the class list file. Each record contains one field for each interest topic. The field contains a coverage rating for that topic by the class as follows:

1 - no relevance
2 - slightly relevant
3 - fairly relevant
4 - highly relevant

12. interest topic description file
- this file is used only by ADVISOR and contains the number of interest topics in the preceding file and provides a title
for each topic for report purposes. Again, its sequence is identical to the class list file.

4.5 The INFORMER Program

4.5.1 General Overview

The need for an improved information system was emphasized in the Introduction of this thesis. The type of information required was discussed resulting in a list of suggested information categories that the ideal course-advising system should make available to the student. INFORMER has been designed as an information retrieval program to provide this class information upon demand.

INFORMER is an independent program and is not activated by any other programs in the system. The student may use INFORMER with or without using ADVISOR. The method of using INFORMER in the pilot study is similar to ADVISOR in that the student initiates his request for class information by filling out a form specifying his name, student number and desired information. The forms are submitted to a central location where they are subsequently keypunched and submitted with a number of other student requests in a batch to INFORMER. The output is in printed form and is manually separated and then distributed to the students.

The student must specify for each class what types of information he would like. The following categories can be retrieved individually for each class:
1) description of class contents  
   - includes past performance of students
2) objectives of the class
3) organization of the class
4) major topics of the class
5) sections, times, rooms and instructors for the class

The system will also provide a list of all classes in the department offered at a certain day and time specified by the student. The purpose of this option is to help a student determine what classes are available for electives that fit his timetable.

4.5.2 INFORMER Logic Narrative

The initial task of INFORMER is to read the entire time index file and class list file from disk into core tables. The information contained by these two files remain in the tables to serve the processing of requests for all students in the batch. The remaining files are accessed by INFORMER only as required and on a direct basis.

INFORMER then reads in the first student number and name as well as a code which indicates whether the student wishes information on specific classes or that he only wants to use the inverted timetable portion of INFORMER.

If the student is requesting information on specific classes, he has indicated the class name (such as CMPT 177) and the option codes indicating the information he would like for that class. The codes are
as follows:

1 = class description
2 = class objectives
3 = class organization
4 = class major topics
5 = sections, times, rooms and instructors

The presence of any of the above five codes triggers a file access and subsequent report printing. The file is accessed using the sequence number determined by searching the class list table for the class in question. The option code in effect designates which file will be accessed (i.e., 1 = class description file, 2 = class objectives file, etc.).

After all of the option codes have been processed and the information printed for a class, INFORMER proceeds to process the next class and its information option codes requested by the student (a maximum of ten classes per student).

At the conclusion of the processing of the specific class requested by the student, INFORMER determines whether the student wishes to use the inverted timetable portion. If not, the program goes back to read the next student request. If the student does want to know what classes are offered on a certain day and time, he must supply the correct codes for the appropriate day, starting times and term.

INFORMER uses the same algorithm to locate the record in the inverted time file as did ADLIB when it created the file. The contents of a record in the inverted time file are the sequence numbers of those
classes that are offered on the specified day, time and term. INFORMER uses the sequence numbers contained in the record to locate the class names in the class list table and then prints the report listing the classes offered at the requested times.

INFORMER then determines if the student has more requests for this function (a maximum of ten per student is allowed), and if not, returns to process the next student.

A source program listing of INFORMER is presented in Appendix B.

4.5.3 INFORMER Example

An example of the forms used for INFORMER and the resulting output is presented on the following few pages.

The first page of the form contains the student name and number and instructions on what information can be retrieved. PART A contains the request for specific information on certain classes. In the example, the student wanted all five categories of information for each of the five classes.

On the second page, PART B is used for requesting a listing of all classes offered at a certain time, day and term. The correct codes are presented for all possible starting times and days. In this example, this student has first of all asked for a listing of all classes on Monday, Wednesday and Friday, starting at 9:30 A.M. for the "B" term.

INFORMER's output is printed on the next seven pages. The first page contains the student name and number and is used mainly for
separating output for each student. The second page contains the output for each of the categories requested for the first class. On the next page, a message has appeared that CMPT 214 was not found in this department's class list. This is a result of CMPT 214 being replaced by another class.

The last page is a printout of the PART B requests and is self-explanatory.
This system can be used to retrieve two types of information:

PART A: INFORMATION ABOUT CERTAIN CLASSES

1. Description and prerequisites of the class and past performance of students.
2. Objectives of the class.
3. Class Organization and Workload.
4. Major Topics of the Class.
5. Sections, times, rooms and instructors of the class.

PART B: WHAT CLASSES ARE OFFERED AT CERTAIN TIMES.

PART A: INFORMATION ABOUT CERTAIN CLASSES.

1. Do you want any of this information (YES or NO) **YES**

2. List classes (maximum of 10) and options wanted (from above).
(Example: CMPT 380  1 2 3 4 5)

<table>
<thead>
<tr>
<th>Class</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 211A</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>CMPT 214B</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>CMPT 212A</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>CMPT 371A</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>CMPT 378B</td>
<td>1 2 3 4 5</td>
</tr>
</tbody>
</table>
PART B: CLASSES OFFERED AT CERTAIN TIMES.

1. Do you want to use this section (YES or NO) **YES**

<table>
<thead>
<tr>
<th>CODES</th>
<th>DAY</th>
<th>STARTING TIME</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Monday</td>
<td>0830</td>
<td>1. Full Class</td>
</tr>
<tr>
<td>2.</td>
<td>Tuesday</td>
<td>0900</td>
<td>2. &quot;A&quot; Term</td>
</tr>
<tr>
<td>3.</td>
<td>Wednesday</td>
<td>0930</td>
<td>3. &quot;B&quot; Term</td>
</tr>
<tr>
<td>4.</td>
<td>Thursday</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Friday</td>
<td>1030</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>M.W.F.</td>
<td>1100</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>T.Th.S.</td>
<td>1130</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>M.W.</td>
<td>1200</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>T.Th.</td>
<td>1230</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td></td>
<td>0100</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td></td>
<td>0130</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td></td>
<td>0200</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td></td>
<td>0230</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td></td>
<td>0300</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td></td>
<td>0330</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td></td>
<td>0400</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td></td>
<td>0430</td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td></td>
<td>0700</td>
<td></td>
</tr>
</tbody>
</table>

2. Fill in the codes in the appropriate slots for the day, time and term for each request you have (maximum of 10). (Example: if you want to know all classes offered by this department at 10:30 on Monday, Wednesday, and Friday starting in January, the right codes would be:

```
6 5 2
```

<table>
<thead>
<tr>
<th>DAY</th>
<th>TIME</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>691662</td>
<td>ROBERT TWYVER</td>
</tr>
<tr>
<td>INFORMER</td>
<td>571662</td>
<td>ROBERT TWYVER</td>
</tr>
</tbody>
</table>
CMPT 211A/D INTRODUCTION TO COMPUTER SCIENCE

CLASS DESCRIPTION:
AN INTRODUCTION TO THE BASIC CONCEPTS OF COMPUTATIONAL SCIENCES, PROPERTIES OF ALGORITHMS, ANALYSIS AND SOLUTION OF COMPUTATIONAL PROBLEMS, INSTRUCTION AND EXPERIENCE IN THE USE OF FORTRAN. THIS CLASS IS INTENDED FOR STUDENTS IN THE PHYSICAL SCIENCE AND IN ENGINEERING.
PAST PERFORMANCE OF STUDENTS (IN PER CENT): A - 30  B - 23  C - 17  D - 10  F - 16
PREREQUISITE: MATHEMATICS 179.

CMPT 211A AND D INTRODUCTION TO COMPUTER SCIENCE

CLASS OBJECTIVES:
THIS COURSE IS DESIGNED TO PROVIDE AN INTRODUCTION TO THE APPLICATION OF COMPUTERS FOR NUMERICAL PROBLEM SOLVING. THE LANGUAGE USED IS MATHEW AND THE EXAMPLES USED ARE ORIENTED TOWARD ENGINEERING TOPICS.

CMPT 211A/J INTRODUCTION TO COMPUTER SCIENCE

CLASS ORGANIZATION:
THEORY EXAMPLE PROBLEMS ARE DEVELOPED IN LECTURES. STUDENTS ARE EXPECTED TO COMPLETE APPROXIMATELY 10 PROGRAMMING ASSIGNMENTS IN THE LABORATORY.

CMPT 211A/3 INTRODUCTION TO COMPUTER SCIENCE

MAJOR TOPICS:
FORTRAN PROGRAMMING;
NUMERICAL INTEGRATION;
SORTING TECHNIQUES;
SIMULATIONS;
ASSEMBLER LANGUAGE PROGRAMMING.

SECTIONS AND INSTRUCTORS FOR CMPT 211

FA'72 Semester Sec 23 N 7:30-10:30PM ARTS 258 E. LOVE
CMPT 212A/B INTRODUCTION TO SYSTEMS PROGRAMMING

CLASS OBJECTIVES:
Systems analysts and system programmers require a knowledge of all levels of software in order to design, use and maintain modern computing systems. The course provides an introduction to machine language programming and the basic hardware components of the computer. It also provides programming experience with an assembly language. The basic characteristics of input-output devices and operation systems are also discussed.

CLASS ORGANIZATION:
This course assumes a knowledge of FORTRAN, BASIC or another similar high-level language as well as an understanding of data structures (e.g., lists, trees, other radix, arithmetic, etc.) A programming lab supplements the class.

SECTIONS AND INSTRUCTORS FOR CMPT 212

Fall Term: Sec 1 M.W. 9:35-10:50 Comm 143 T. Volk

Fall Term: Sec 2 M.W. 8:30-9:30 Comm 162 J. Cole
CMPT 377A/B DATA PROCESSING

CLASS OBJECTIVES:

This course is divided into two parts. The first part is concerned with presenting the COBOL language at the same time placing emphasis on programming techniques useful in typical data processing applications. The second part concentrates on business systems analysis. A methodical approach to solving problems in this area is presented, drawing many examples from real life situations. The student should accomplish the following during the course: develop a skill at COBOL programming by introducing to file handling methods and associated problems; become familiar with the systems analysis approach to problems; and receive some insight into the problems of systems design.

CMPT 377A OR B DATA PROCESSING

CLASS ORGANIZATION:

Initially the course follows a lecture-problem solving format as the COBOL language is taught and students solve business problems in COBOL using the university computer. As the class moves into systems analysis and design lectures center around the discussion of these techniques and case studies of business applications. Although a formal lab is not offered, students should be prepared for extensive programming, analysis and design problem assignments.

CMPT 377A/B DATA PROCESSING

MAJOR TOPICS:

COBOL PROGRAMMING

SYSTEMS ANALYSIS

SYSTEMS DESIGN

SYSTEMS IMPLEMENTATION.

SECTIONS AND INSTRUCTORS FOR CMPT 377

FA TERM  SEC 51  MWF  10:00-10:50  COMM 44  W. PASK

SEC 09  Tu.  7:00-11:30  COMM 245  W. GLASEL

FA TERM  SEC 51  MWF  10:00-11:00  COMM 44  W. PASK

SEC 19  Tu.  7:00-11:30  COMM 245  W. GLASEL
CMPT 378A SIMULATION PRINCIPLES I

CLASS DESCRIPTIONS:
Introduction to the development of logical numerical and statistical models of systems; deterministic and probabilistic models; Monte Carlo simulations; basic elements involved in simulation such as entities, attributes, events; and the representation of properties of computer modeling languages and models. Development and testing of models; past performance of students in percentiles.

A - 12, B - 10, C - 8, D - 6, F - 0

PREREQUISITES: CMPT 181, 211, MAT 241, 242, or permission of the instructor.

CMPT 378A SIMULATION PRINCIPLES I

CLASS OBJECTIVES:
The class is an introductory class on the principles of model building using computers. The objectives are as follows: to explain the concepts behind the use of computer simulation; to teach and practice the tools and techniques needed to build models; to discuss the steps involved and problems encountered in building practical complex models, and to survey past, present, and future uses of computer simulation.

CMPT 378A SIMULATION PRINCIPLES I

CLASS ORGANIZATION:
The required text is a detailed manual of notes prepared by the class instructors. The lectures summarize the more important parts of these notes and serve as a means of clarifying problems. Six programming problems are set with difficulty and complexity increasing. A knowledge of FORTRAN is essential, and use is made of EASYFIR, a set of FORTRAN SUBROUTINES that acts as a simple simulation language.

CMPT 378A SIMULATION PRINCIPLES I

MAJOR TOPICS:
Concepts of simulation; simulation techniques; operational models; applications.

SECTIONS AND INSTRUCTORS FOR CMPT 378

FA'85 TERM
SEC 51, T, TH, 9-10 AM, COMM 245, J. Cooke
SEC 52, TH, 1-2 PM, COMM 245, J. Shanks
<table>
<thead>
<tr>
<th>CLASSES OFFERED BY THIS DEPARTMENT AT SPECIFIED TIMES</th>
</tr>
</thead>
</table>

| CLASSES OFFERED ON MWF. AT 0930 DURING "B" TERM |

| CMPT 181 |
| CMPT 428 |

| CLASSES OFFERED ON MWF. AT 1230 DURING "B" TERM |

| CMPT 177 |
| CMPT 377 |

| CLASSES OFFERED ON TTH. AT 1230 DURING "A" TERM |

| "J" CLASSES OFFERED BY THIS DEPARTMENT AT THIS TIME |
4.6 The ADVISOR Program

4.6.1 General Overview

ADVISOR is the name of the program that performs the advising function of the pilot system. The purpose of ADVISOR is to suggest a set of classes for the student to consider registering in, and informing the student as to the reasons why each class has been suggested. ADVISOR also provides section times and instructors of the suggested classes to aid the student in constructing his class timetable.

ADVISOR has been designed to advise students interested in the Department of Computational Science. The program has been written to address course requirements at the department (or major) level. With very minor changes, ADVISOR could function for the Department of Mathematics, Economics, or Political Science for example.

In order for ADVISOR to perform its function, it must receive at the outset the following information about the student:

- name and student number
- college
- whether major is Computer Science or not
- successfully completed classes

Student records were not available to ADVISOR for the pilot study and as a result, ADVISOR relied upon the student to supply an accurate list of classes that he has successfully completed. In addition to the above, ADVISOR requires some means of determining an indication of the
student's interest in a number of classes. A list of topics is presented to the student and he is requested to indicate his level of interest in each topic as follows:

1 hate
2 no interest
3 slight interest
4 fair interest
5 high interest

In the pilot study, forms were designed for the student to provide the necessary information. The completed forms were subsequently key-punched and submitted in a batch with requests from other students to ADVISOR for processing.

ADVISOR produces output in the form of printed reports identifying:

- the student's name, student number, college and whether a computer science major or not
- the classes taken by the student (as supplied by the student)
- the required classes for a major (not printed for non-major students)
- interest topics and the student's ratings
- instructions for the use of ADVISOR and how to read the schedules
- two schedules of suggested classes. Schedule I is for students planning to take computer science classes in
subsequent years, and Schedule II is for students who will be taking them in the upcoming term only.

times, rooms and instructors for each section of suggested classes for both schedules

4.6.2 ADVISOR Logic Narrative

The intent of this section is to describe how ADVISOR performs the more important aspects of its function.

In suggesting a set of classes, ADVISOR must take the following factors into consideration:

1. classes already taken
2. required classes
3. required options
4. classes of interest to the student
5. prerequisites of the above three types of classes

The first task is to identify the potential classes. A potential class is one which may be taken in the term for which the student is registering. A potential full class is one whose prerequisites have been satisfied in previous years. The same is true for a potential "A" half class. However, a potential "B" half class would exist if the prerequisites were taken in the previous years or in the first half of the coming year.

Each class has a status code associated with it. Examining the status code will tell whether the class:
1 has been taken
2 is a potential full class
3 is a potential "A" half class
4 is a potential "B" half class
5 is a potential "A" or "B" half class
6 a potential "B" half class with a potential "A" half class as a prerequisite

The taken classes (as submitted by the student) are compared to the class list file and for those classes where a match is found, the status code is set to 1.

The procedure for determining whether a class is a potential full or "A" half class is to first of all locate (by the use of the class type code in the class list file) one of these classes in the class list file and then to check the prerequisite file to see if there are any prerequisites. If not, the status code can be updated accordingly. If prerequisites do exist, then their status codes can be checked to see if they have been taken.

Once all the "A" half classes have been processed, the "B" half classes and their prerequisites can be checked and the status code updated.

The net result of the above procedure is that all the classes in the class list file will have an updated status code with the exception of those classes whose prerequisites had not been satisfied and whose status code remains at zero.
For those students who intend to major in computer science, it is necessary to consider what required classes can be taken. The required classes have been previously read into a table for his college from the required classes file. Only the required classes are considered at this time and not those where the student is required to choose a certain number of classes from a given set of classes. These classes are analyzed after the interest matching procedure is completed.

The status codes of the required classes are examined to see if the class has been taken or has potential status. If the required class has potential status, then it is given a heavy weight to ensure that it will be suggested as a class for the student to take.

However, if the class does not have potential status, its prerequisites will be examined. If any of the prerequisites have potential status, then they will be given a heavy weight to ensure that they are suggested.

The next step is to determine the student's interest levels in those classes that have not been taken or are not required classes. The student has submitted an interest rating for each topic on the following basis:

1 hate
2 no interest
3 slight interest
4 fair interest
5 high interest
The interest rating for each of the topics is compared with the rating given by the instructor for each class (stored in the topic coverage file).

A total interest weight is computed for each class. This weight is comprised of individual weights given for each topic. The weighting scheme that was finally selected as the most appropriate is presented in Table 4.1. This matrix indicates that range of weights can be from -75 to +30. The -75 weight is attained by a class which has a high coverage of a topic that the student "hates." The +30 weight is given to a class that has a high coverage of a topic in which the student has indicated a high interest.

After all classes had been given their total interest weighting, each class's status code is examined to see if the class could be taken this term. If so, then the total interest weight would be added to the class's total weight.

If the class was not potential, and the interest was above average for the student, then the prerequisites are examined and any prerequisites which had potential status would receive the total interest weight added to its total weight. In this manner, a class which the student was very interested in would not be overlooked and its prerequisites would be satisfied enabling him to eventually take the class.

In some cases, in order to fulfill the requirements of a major in a certain field, the student must choose a minimum number of classes out of a given set. These are referred to as "required options." The major criteria for selecting from these classes should be interest.
<table>
<thead>
<tr>
<th>STUDENT</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Hate)</td>
<td>0</td>
<td>-25</td>
<td>-50</td>
<td>-75</td>
</tr>
<tr>
<td>(None)</td>
<td>0</td>
<td>-10</td>
<td>-20</td>
<td>-30</td>
</tr>
<tr>
<td>INTEREST</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>-10</td>
</tr>
<tr>
<td>(Slight)</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>RATING</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>(Fair)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(High)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 4.1

MATRIX OF INTEREST WEIGHTS
The number of classes the student must choose and also the set of classes from which the choice is made is specified in the required classes file. The classes in this set already have a total interest weight associated with them. The highest weighted classes (as many as the student is required to choose) are examined to see if they are potential classes or not. If they are, a high weight is added to their total weight to ensure that they are selected. If they are not potential classes, and have not already been taken, then their prerequisites are examined and weighted in a manner similar to the required classes.

At this phase of the system, all classes have been identified in the following manner:

0 Senior class
1 Taken class
2 Potential full class
3 Potential "A" half class
4 Potential "B" half class
5 Potential "A" or "B" half class
6 Potential "B" half class with a potential "A" half class as a prerequisite

As well, each class will have a total weight comprised of weights assigned for the following:

1. a required class
2. a prerequisite for a senior required class
3. a required option class
4. a prerequisite for a required option
5. an interest weight
6. a prerequisite for a senior class with an above-average interest weight

The system does not attempt to structure a timetable for the student because of the fact that the student has other classes from other colleges that he will have to take. The system does print out a report listing the classes in sequence of highest weight and indicates the sources of the weights as well. The student then uses this as a guide to select the classes.

A source program listing of ADVISOR is presented in Appendix C.

4.6.3 ADVISOR Example

The purpose of this subsection is to give two examples of how ADVISOR performs—one for a student who is majoring in the particular department (in this case Computational Science) and the other for a student whose major is in another field and is taking classes from this department as electives.

4.6.3.1 Example 1—Non-major

The first example to be discussed is the non-major student. The input form as completed by the student are presented on the following three pages.
DEPARTMENT OF COMPUTATIONAL SCIENCE

COURSE ADVISING

NAME  ALEX Mcgregor

STUDENT NUMBER  673530

COLLEGE  Arts + Science

MAJOR  _______________ NONMAJOR  √  IN THIS FIELD

PLEASE LIST CAREFULLY ALL THE QUANT, CMPT AND MATH CLASSES
YOU HAVE SUCCESSFULLY COMPLETED TO DATE

THE CORRECT FORM IS:

Math 102  __________
CMPT 177  __________

_______  __________
_______  __________
_______  __________
_______  __________
_______  __________
_______  __________
_______  __________
_______  __________
INTEREST TOPICS

One of the most important functions of this advising system is to make sure that the student's interests are recognized and taken into account when suggesting classes.

Below you will find a list of interest topics for which you are to give an indication of your interest in each topic. The way you are to do this is to put your interest rating for each topic on the right of the topic in the space allocated for this purpose. The rating is as follows:

1 - NO INTEREST
2 - SLIGHT INTEREST
3 - FAIR INTEREST
4 - HIGH INTEREST

For example, if for the first topic, "STATISTICAL USE OF THE COMPUTER," you think you are very interested in this area, you should rate your interest in that topic as 4.

Please be sure to rate your interest in all topics.

<table>
<thead>
<tr>
<th>Interest Topic</th>
<th>Interest Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. STATISTICAL USE OF THE COMPUTER</td>
<td>2</td>
</tr>
<tr>
<td>2. SIMPLE PROBLEM SOLVING</td>
<td>3</td>
</tr>
<tr>
<td>3. GENERAL USES OF COMPUTERS</td>
<td>3</td>
</tr>
<tr>
<td>4. SCIENTIFIC OR COMPLEX PROBLEM SOLVING</td>
<td>4</td>
</tr>
<tr>
<td>5. INTERNAL DESIGN OF COMPUTERS</td>
<td>3</td>
</tr>
<tr>
<td>6. DEVELOPING BUSINESS COMPUTER SYSTEMS</td>
<td>2</td>
</tr>
<tr>
<td>7. DETAILED ANALYSIS OF A COMPUTER LANGUAGE</td>
<td>3</td>
</tr>
<tr>
<td>8. BASIC CONCEPTS OF COMPILERS</td>
<td>3</td>
</tr>
<tr>
<td>(explanation: A compiler is used to translate a</td>
<td></td>
</tr>
<tr>
<td>language into a form the computer can understand.</td>
<td></td>
</tr>
<tr>
<td>9. MANAGEMENT AND CONTROL OF COMPUTERS</td>
<td>2</td>
</tr>
<tr>
<td>10. OPERATIONS RESEARCH AND MATHEMATICAL PROGRAMMING</td>
<td>1</td>
</tr>
<tr>
<td>(explanation: Operations Research is the solving of managerial problems using mathematics.)</td>
<td></td>
</tr>
<tr>
<td>Interest Topic</td>
<td>Interest Rating</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>11. HEURISTICS AND ARTIFICIAL INTELLIGENCE</td>
<td>2</td>
</tr>
<tr>
<td>(explanation: Heuristics involves the act of</td>
<td></td>
</tr>
<tr>
<td>learning by the computer, and simulation of</td>
<td></td>
</tr>
<tr>
<td>human operations to solve, for example, problems</td>
<td></td>
</tr>
<tr>
<td>of scheduling and routing.)</td>
<td></td>
</tr>
<tr>
<td>12. TELECOMMUNICATIONS THEORY</td>
<td>1</td>
</tr>
<tr>
<td>13. REAL TIME SYSTEMS</td>
<td>2</td>
</tr>
<tr>
<td>(explanation: Real time systems are usually</td>
<td></td>
</tr>
<tr>
<td>very large computer systems and allow instant</td>
<td></td>
</tr>
<tr>
<td>action on the part of the human user or</td>
<td></td>
</tr>
<tr>
<td>mechanical process.)</td>
<td></td>
</tr>
<tr>
<td>14. DESIGN OF LARGE SCALE COMPUTERS</td>
<td>2</td>
</tr>
<tr>
<td>15. MANIPULATION AND STORAGE OF DATA</td>
<td>2</td>
</tr>
<tr>
<td>16. COMPUTER GRAPHICS</td>
<td>4</td>
</tr>
<tr>
<td>(explanation: The displaying of information</td>
<td></td>
</tr>
<tr>
<td>in a graphic form and the dynamic manipulation</td>
<td></td>
</tr>
<tr>
<td>of geometric figures.)</td>
<td></td>
</tr>
<tr>
<td>17. USING THE COMPUTER FOR SIMULATION</td>
<td>3</td>
</tr>
<tr>
<td>18. PRINCIPLES OF COMPUTER LOGIC AND ORGANIZATION</td>
<td>2</td>
</tr>
<tr>
<td>19. FILE HANDLING TECHNIQUES</td>
<td>2</td>
</tr>
<tr>
<td>20. EVALUATION OF COMPUTER SYSTEMS</td>
<td>3</td>
</tr>
<tr>
<td>21. COMBINATORIAL MATHEMATICS</td>
<td>3</td>
</tr>
<tr>
<td>(explanation: Dealing with and manipulation of</td>
<td></td>
</tr>
<tr>
<td>permutations and combinations.)</td>
<td></td>
</tr>
<tr>
<td>22. ANALYSIS OF ALGORITHMS</td>
<td>1</td>
</tr>
<tr>
<td>(explanation: An algorithm is usually a simplified</td>
<td></td>
</tr>
<tr>
<td>procedure for solving a complex problem.)</td>
<td></td>
</tr>
<tr>
<td>23. THEORY OF AUTOMATA</td>
<td>2</td>
</tr>
<tr>
<td>(explanation: Mathematical ideas on which</td>
<td></td>
</tr>
<tr>
<td>computer science is based.)</td>
<td></td>
</tr>
<tr>
<td>24. FORMAL LANGUAGES</td>
<td>3</td>
</tr>
<tr>
<td>(explanation: Properties and representation of</td>
<td></td>
</tr>
<tr>
<td>computer languages.)</td>
<td></td>
</tr>
</tbody>
</table>
The first page gives the student name, number, college and whether majoring in Computer Science or not. The student is asked to list all of the relevant classes that he has successfully completed to date.

The second and third pages of the form contain the interest topics and the student's interest rating. It should be noted that in the example given there are only four ratings to choose from. The fifth rating "hate" was added at a later stage of the pilot study.

The output from ADVISOR for this student is presented on the next eight pages.
COURSE ADVISING FOR THE DEPARTMENT OF COMPUTATIONAL SCIENCE

ALEX W. MCGREGOR  673530
COLLEGE OF COMMERCE

THIS STUDENT IS NOT MAJORING IN THIS FIELD
COURSE ADVISING PORTION

ALEX W. MCGROR  673330

CLASSES TAKEN BY THIS STUDENT

MATH 102
CMPE 177
## INTEREST TOPICS AND STUDENT RATINGS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Use of the Computer</td>
<td>2</td>
</tr>
<tr>
<td>Simple Problem Solving</td>
<td>3</td>
</tr>
<tr>
<td>General Uses of Computers</td>
<td>3</td>
</tr>
<tr>
<td>Scientific or Complex Problem Solving</td>
<td>4</td>
</tr>
<tr>
<td>Internal Design of Computers</td>
<td>3</td>
</tr>
<tr>
<td>Developing Business Computer Systems</td>
<td>2</td>
</tr>
<tr>
<td>Detailed Analysis of a Computer Language</td>
<td>3</td>
</tr>
<tr>
<td>Basic Concepts of Compilers</td>
<td>3</td>
</tr>
<tr>
<td>Management and Control of Computers</td>
<td>2</td>
</tr>
<tr>
<td>Operations Research and Mathematical Programming</td>
<td>1</td>
</tr>
<tr>
<td>Heuristics and Artificial Intelligence</td>
<td>2</td>
</tr>
<tr>
<td>Telecommunications Theory</td>
<td>1</td>
</tr>
<tr>
<td>Real Time Systems</td>
<td>2</td>
</tr>
<tr>
<td>Design of Large Scale Computers</td>
<td>2</td>
</tr>
<tr>
<td>Manipulation and Storage of Data</td>
<td>2</td>
</tr>
<tr>
<td>Computer Graphics</td>
<td>4</td>
</tr>
<tr>
<td>Using the Computer for Simulation</td>
<td>3</td>
</tr>
<tr>
<td>Principles of Computer Logic and Organization</td>
<td>2</td>
</tr>
<tr>
<td>File Handling Techniques</td>
<td>2</td>
</tr>
<tr>
<td>Evaluation of Computer Systems</td>
<td>3</td>
</tr>
<tr>
<td>Combinatorial Mathematics</td>
<td>3</td>
</tr>
<tr>
<td>Analysis of Algorithms</td>
<td>1</td>
</tr>
<tr>
<td>Theory of Automata</td>
<td>2</td>
</tr>
<tr>
<td>Formal Languages</td>
<td>3</td>
</tr>
</tbody>
</table>
INSTRUCTIONS FOR USING ADVISOR

ON THE FOLLOWING PAGES ARE TWO SCHEDULES OF SUGGESTED CLASSES THAT YOU CAN POSSIBLY TAKE NEXT YEAR. FROM THE APPROPRIATE SCHEDULE YOU ARE TO SELECT THE CLASSES YOU WISH TO TAKE FROM THIS DEPARTMENT FOR THE COMING YEAR. THE CLASSES ARE ORDERED BY THEIR FINAL RATING, WITH THE LARGEST FIRST, AND THE REASONS FOR THE RATING ARE SHOWN FOR EACH CLASS.

WHEN SELECTING CLASSES FROM THIS LIST TO FIT YOUR TIMETABLE, IT IS IMPORTANT THAT YOU ATTEMPT TO TAKE THOSE CLASSES FROM THE TOP OF THE LIST. FOR EXAMPLE, IF YOU WISH TO TAKE FOUR CLASSES FROM THIS DEPARTMENT, YOU SHOULD SELECT THE TOP FOUR IN THE SCHEDULE. IF HOWEVER, ONE OF THESE CLASSES CANNOT FIT INTO YOUR TIMETABLE, THEN YOU SHOULD SUBSTITUTE THE FIFTH HIGHEST RATED CLASS IN THE LIST, AND SO ON.

SCHEDULE 1

THIS SET OF CLASSES IS MEANT FOR THE STUDENT WHO WILL BE TAKING ADDITIONAL CLASSES FROM THIS DEPARTMENT AFTER THE COMING YEAR. IT IS IMPORTANT THEREFORE THAT THE PREREQUISITES FOR SENIOR CLASSES ARE SATISFIED NEXT YEAR, SO THAT THE SENIOR CLASSES MAY BE TAKEN IN THE FOLLOWING YEAR.

THIS SCHEDULE, THEREFORE, GIVES A HIGHER RATING TO THOSE CLASSES THAT ARE PREREQUISITES FOR:

1. A REQUIRED CLASS, IN THE CASE OF A MAJOR,

2. A REQUIRED OPTION, IN THE CASE OF A MAJOR

3. A SENIOR CLASS IN WHICH THE STUDENT HAS AN ABOVE AVERAGE INTEREST

SCHEDULE 2

THIS SET OF CLASSES IS MEANT FOR THE STUDENT WHO WILL NOT BE TAKING ADDITIONAL CLASSES FROM THIS DEPARTMENT AFTER THE COMING YEAR. THEREFORE, ONLY THOSE CLASSES THAT CAN BE TAKEN IN THE COMING YEAR WILL BE CONSIDERED.
**Schedule 1: If you are planning to take additional classes after the coming year**

**Suggested Classes for Alex W. McGregor 673530**

<table>
<thead>
<tr>
<th>Class</th>
<th>Final Rating</th>
<th>Student Interest Rating</th>
<th>Prereq for Senior Class with Above Avg Interest Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 212 AGRB</td>
<td>210*</td>
<td>94</td>
<td>X</td>
</tr>
<tr>
<td>CMPT 181 B</td>
<td>200*</td>
<td>50</td>
<td>X</td>
</tr>
<tr>
<td>NOTE 1. CMPT 316 B</td>
<td>160*</td>
<td>90</td>
<td>X</td>
</tr>
<tr>
<td>CMPT 215 A</td>
<td>110*</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>CMPT 176 B</td>
<td>80*</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>CMPT 377 AGRB</td>
<td>50*</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE 1. The following A half classes must be taken before this A half class CMPT 215**

**Times and instructors of the top ten ranking classes from above are given on the following page**

---

**Note:**

- CMPT 212 AGRB
- CMPT 181 B
- CMPT 316 B
- CMPT 215 A
- CMPT 176 B
- CMPT 377 AGRB
- CMPT 476 B

---

**Reasons for being suggested**

- X
- ✔

---

Additional notes or comments:

- The following A half classes must be taken before this A half class CMPT 215.
- Times and instructors of the top ten ranking classes from above are given on the following page.
TIMES AND INSTRUCTORS OF TOP TEN CLASSES

CMPT 212 A0R B  INTRODUCTION TO SYSTEMS PROGRAMMING
  "A" TERM SEC 01 M.W.F. 9:30-10:30 COMM 142 J. VOLK
  "B" TERM SEC 01 M.W.F. 8:30-9:30 COMM 142 D. COLE

CMPT 191 B  INTRODUCTION TO DATA STRUCTURES AND NONNUMERIC COMPUTING
  "A" TERM SEC 01 M.W.F. 9:30-10:30 COMM 144 J. TREMBLAY
  "B" TERM SEC 01 M.W.F. 9:30-10:30 ARTS 258 B. HOLMLUND
  SEC 02 T.R.F. 11:00-11:30 COMM 144 B. HOLMLUND

CMPT 316 B  COMPILER CONSTRUCTION
  "A" TERM SEC 01 M.W.F. 8:30-9:30 ARTS 258 R. BOOTH

CMPT 215 A  COMPUTER PROGRAMMING LANGUAGES
  "A" TERM SEC 01 M.W.F. 10:30-11:30 ARTS 258 R. BOOTH

CMPT 170 B  INTRODUCTION TO COMPUTERS II
  "B" TERM SEC 01 M.W.F. 11:30-12:30 COMM 145 M. PASK
  SEC 02 M.W.F. 8:30-9:30 COMM 145 K. VOLK
  SEC 03 T.R.F. 11:00-11:30 COMM 144 M. PASK
  SEC 09 M.W. 7:00-9:00PM COMM 262 C. STOCK

CMPT 377 A0R B  DATA PROCESSING
  "A" TERM SEC 01 M.W.F. 10:30-11:30 COMM 144 M. PASK
  "B" TERM SEC 01 M.W.F. 10:30-11:30 COMM 144 M. PASK
  SEC 09 T. 7:00-10:00PM COMM 265 N. GLASSEL

CMPT 476 B  REAL-TIME INFORMATION SYSTEMS
  "B" TERM SEC 01 M.W. 11:30-12:00 COMM 142 G. METER
SCHEDULE 2. IF YOU ARE NOT PLANNING TO TAKE ADDITIONAL CLASSES AFTER THE COMING YEAR

SUGGESTED CLASSES FOR ALEX W. MCGREGOR 473530

<table>
<thead>
<tr>
<th>CLASS</th>
<th>FINAL RATING</th>
<th>INTEREST RATING</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 215 A</td>
<td>110*</td>
<td>110</td>
</tr>
<tr>
<td>NOTES CMPT 316 B</td>
<td>90*</td>
<td>90</td>
</tr>
<tr>
<td>CMPT 212 AORB</td>
<td>90*</td>
<td>90</td>
</tr>
<tr>
<td>CMPT 178 B</td>
<td>80*</td>
<td>80</td>
</tr>
<tr>
<td>CMPT 181 B</td>
<td>50*</td>
<td>50</td>
</tr>
<tr>
<td>CMPT 377 AORB</td>
<td>50*</td>
<td>50</td>
</tr>
<tr>
<td>CMPT 476 B</td>
<td>50*</td>
<td>50</td>
</tr>
</tbody>
</table>

NOTE: THE FOLLOWING A HALF CLASSES MUST BE TAKEN BEFORE THIS B HALF CLASS CMPT 213

TIMES AND INSTRUCTORS OF THE TOP TEN RANKING CLASSES FROM ABOVE ARE GIVEN ON THE FOLLOWING PAGE
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Term</th>
<th>Sec</th>
<th>Days</th>
<th>Time</th>
<th>Location</th>
<th>Instructor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 215 A</td>
<td>COMPUTER PROGRAMMING LANGUAGES</td>
<td>&quot;A&quot;</td>
<td>01</td>
<td>M,W,F</td>
<td>9:30-</td>
<td>ARTS 258</td>
<td>K. BOOTH</td>
<td></td>
</tr>
<tr>
<td>CMPT 316 B</td>
<td>COMPILER CONSTRUCTION</td>
<td>&quot;A&quot;</td>
<td>01</td>
<td>M,W,F</td>
<td>9:30-</td>
<td>ARTS 258</td>
<td>K. BOOTH</td>
<td></td>
</tr>
<tr>
<td>CMPT 212 AQRB</td>
<td>INTRODUCTION TO SYSTEMS PROGRAMMING</td>
<td>&quot;A&quot;</td>
<td>01</td>
<td>M,W,F</td>
<td>9:10-</td>
<td>COMM 143</td>
<td>L. VOLK</td>
<td></td>
</tr>
<tr>
<td>CMPT 170 B</td>
<td>INTRODUCTION TO COMPUTERS II</td>
<td>&quot;B&quot;</td>
<td>01</td>
<td>T,TH</td>
<td>1:10-</td>
<td>COMM 144</td>
<td>C. STOCK</td>
<td></td>
</tr>
<tr>
<td>CMPT 181 B</td>
<td>INTRODUCTION TO DATA STRUCTURES AND NONNUMERIC COMPUTING</td>
<td>&quot;B&quot;</td>
<td>01</td>
<td>T,TH</td>
<td>1:10-</td>
<td>COMM 144</td>
<td>B. HOLLUND</td>
<td></td>
</tr>
<tr>
<td>CMPT 377 AQRB</td>
<td>DATA PROCESSING</td>
<td>&quot;A&quot;</td>
<td>01</td>
<td>T,TH</td>
<td>10:10-</td>
<td>COMM 144</td>
<td>M. GLASSEL</td>
<td></td>
</tr>
<tr>
<td>CMPT 476 B</td>
<td>REAL-TIME INFORMATION SYSTEMS</td>
<td>&quot;B&quot;</td>
<td>01</td>
<td>T,TH</td>
<td>10:10-</td>
<td>COMM 142</td>
<td>C. HEATER</td>
<td></td>
</tr>
</tbody>
</table>
The first page shows the student name, number, college and whether a major or not.

The second page lists the relevant classes successfully completed as submitted by the student. The third page lists the student's interest ratings for each topic.

The fourth page contains instructions for the use of ADVISOR and how to read the schedules.

The fifth page contains Schedule 1. This schedule should be considered by students considering taking more classes from the department in subsequent years.

Schedule 1 contains a list of only potential classes in order of their total weights. In the case of a non-major, the weighting is comprised of an interest rating for the class, plus the interest weight of a senior class (with an above-average interest rating) for which the class is a prerequisite. In the example, CMPT 212 A or B has a final rating of 210. This is comprised of an interest rating of 90 on its own, plus an interest weight of 120 from a class for which CMPT 212 is a prerequisite. Incidentally, it was later suggested by students involved in the pilot study that it would be helpful if the senior class was identified.

According to Schedule 1 there are seven computer science classes for which the student has satisfied the prerequisites. The student is instructed to choose from the top of the list of Schedule 1 if he plans to take more computer science classes in years to come.
To aid the student in setting up his timetable, the sections, times, rooms and instructors of the suggested classes are listed on the next page of the printout.

Schedule 2 is intended to be used for students who do not plan to take classes from the department in any other years. As a result, the interest levels of senior classes are ignored and the classes are ranked in order of their interest rating.

It should be noted that in the example the positions of the classes in the schedule have changed significantly from Schedule 1.

4.6.3.2 Example 2--Major

The second example consists of ADVISOR's output for a student who is majoring in Computational Science. For reasons of brevity, the input forms are not presented in this example. The same forms as in the first example were used. ADVISOR's output is presented on the following nine pages.
COURSE ADVISING FOR THE DEPARTMENT OF COMPUTATIONAL SCIENCE

DON SCHELLENBERG  693716
COLLEGE OF ARTS AND SCIENCE

THIS STUDENT IS MAJORING IN THIS FIELD
REQUIREMENTS FOR A MAJOR IN COMPUTATIONAL SCIENCE
FOR THE COLLEGE OF ARTS AND SCIENCE

REQUIRED CLASSES

CMPT 180 A  INTRODUCTION TO COMPUTATIONAL SCIENCE AND ALGORITHMIC PROCESSING
CMPT 212 A  INTRODUCTION TO DATA STRUCTURES AND NONNUMERIC COMPUTING
CMPT 215 A  INTRODUCTION TO SYSTEMS PROGRAMMING
CMPT 220 A  COMPUTER PROGRAMMING LANGUAGES
CMPT 320 A  PRINCIPLES OF LOGIC SYSTEMS
MATH 175 A  ELEMENTARY CALCULUS AND ANALYTICAL GEOMETRY
MATH 211 A  NUMERICAL ANALYSIS

PLUS 2 OF THE FOLLOWING CLASSES

CMPT 377 A  DATA PROCESSING
CMPT 378 A  SIMULATION PRINCIPLES I
CMPT 313 B  INTERMEDIATE SYSTEMS PROGRAMMING
CMPT 316 B  COMPILER CONSTRUCTION
CMPT 393 B  OPERATIONS RESEARCH I

NOTE: THE STUDENT IS ADVISED TO CONSULT THE CALENDAR TO CHECK IF ADDITIONAL
CLASSES MUST BE TAKEN TO COMPLETE MAJOR REQUIREMENTS.
COURSE ADVISING PORTION

DUN. SCHELLENBERG 693716

CLASSES TAKEN BY THIS STUDENT

MATH 102
MATH 270
MATH 222
MATH 260
CMPT 211
CMPT 377

SPECIAL PERMISSION HAS BEEN OBTAINED TO TAKE THE FOLLOWING CLASSES:

CMPT 378
<table>
<thead>
<tr>
<th>Interest Topics</th>
<th>This Student Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Use of the Computer</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Simple Problem Solving</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>General Uses of Computers</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Scientific or Complex Problem Solving</td>
<td>5 <strong>High Interest</strong></td>
</tr>
<tr>
<td>Internal Design of Computers</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Developing Business Computer Systems</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Detailed Analysis of a Computer Language</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Basic Concepts of Compilers</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Management and Control of Computers</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Operations Research and Mathematical Programming</td>
<td>5 <strong>High Interest</strong></td>
</tr>
<tr>
<td>Heuristics and Artificial Intelligence</td>
<td>5 <strong>High Interest</strong></td>
</tr>
<tr>
<td>Telecommunications Theory</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Real Time Systems</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Design of Large Scale Computers</td>
<td>2 <strong>No Interest</strong></td>
</tr>
<tr>
<td>Manipulation and Storage of Data</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Computer Graphics</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Using the Computer for Simulation</td>
<td>5 <strong>High Interest</strong></td>
</tr>
<tr>
<td>Principles of Computer Logic and Organization</td>
<td>5 <strong>High Interest</strong></td>
</tr>
<tr>
<td>File Handling Techniques</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Evaluation of Computer Systems</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Combinational Mathematics</td>
<td>3 <strong>Some Interest</strong></td>
</tr>
<tr>
<td>Analysis of Algorithms</td>
<td>3 <strong>High Interest</strong></td>
</tr>
<tr>
<td>Theory of Automata</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
<tr>
<td>Formal Languages</td>
<td>4 <strong>Fair Interest</strong></td>
</tr>
</tbody>
</table>
INSTRUCTIONS FOR USING ADVISOR

ON THE FOLLOWING PAGES ARE TWO SCHEDULES OF SUGGESTED CLASSES THAT YOU CAN POSSIBLY TAKE NEXT YEAR. FROM THE APPROPRIATE SCHEDULE YOU ARE TO SELECT THE CLASSES YOU WISH TO TAKE FROM THIS DEPARTMENT FOR THE COMING YEAR. THE CLASSES ARE ORDERED BY THEIR FINAL RATING, WITH THE LARGEST FIRST, AND THE REASONS FOR THE RATING ARE SHOWN FOR EACH CLASS.

WHEN SELECTING CLASSES FROM THIS LIST TO FIT YOUR TIMETABLE, IT IS IMPORTANT THAT YOU ATTEMPT TO TAKE THOSE CLASSES FROM THE TOP OF THE LIST. FOR EXAMPLE: IF YOU WISH TO TAKE FOUR CLASSES FROM THIS DEPARTMENT, YOU SHOULD SELECT THE TOP FOUR IN THE SCHEDULE. IF HOWEVER, ONE OR THESE CLASSES CANNOT FIT INTO YOUR TIMETABLE, THEN YOU SHOULD SUBSTITUTE THE FIFTH HIGHEST RATED CLASS IN THE LIST, AND SO ON.

SCHEDULE 1

THIS SET OF CLASSES IS MEANT FOR THE STUDENT WHO WILL BE TAKING ADDITIONAL CLASSES FROM THIS DEPARTMENT AFTER THE COMING YEAR.

IT IS IMPORTANT THAT THE PREREQUISITES FOR SENIOR CLASSES ARE SATISFIED NEXT YEAR SO THAT THE SENIOR CLASSES MAY BE TAKEN IN THE FOLLOWING YEAR.

THIS SCHEDULE, THEREFORE, GIVES A HIGHER RATING TO THOSE CLASSES THAT ARE PREREQUISITES FOR:

1. A REQUIRED CLASS, IN THE CASE OF A MAJOR;
2. A REQUIRED OPTION, IN THE CASE OF A MAJOR;
3. A SENIOR CLASS IN WHICH THE STUDENT HAS AN ABOVE AVERAGE INTEREST.

SCHEDULE 2

THIS SET OF CLASSES IS MEANT FOR THE STUDENT WHO WILL NOT BE TAKING ADDITIONAL CLASSES FROM THIS DEPARTMENT AFTER THE COMING YEAR.

THerefore, only those classes that can be taken in the coming year will be considered.
**SCHEDULE 1. IF YOU ARE PLANNING TO TAKE ADDITIONAL CLASSES AFTER THE COMING YEAR**

**SUGGESTED CLASSES FOR DON SCHELLENBERG 693716**

<table>
<thead>
<tr>
<th>REASONS FOR BEING SUGGESTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS</td>
</tr>
<tr>
<td>CMPT 181 A</td>
</tr>
<tr>
<td>CMPT 212 DURB</td>
</tr>
<tr>
<td>CMPT 215 DURB</td>
</tr>
<tr>
<td>CMPT 378 A</td>
</tr>
<tr>
<td>MATH 311 A</td>
</tr>
<tr>
<td>NOTE 1 CMPT 479 B</td>
</tr>
<tr>
<td>CMPT 476 B</td>
</tr>
<tr>
<td>CMPT 380 B</td>
</tr>
</tbody>
</table>

**NOTE: THE FOLLOWING A HALF CLASSES MUST BE TAKEN BEFORE THIS B HALF CLASS CMPT 378**

**TIMES AND INSTRUCTORS OF THE TOP TEN RANKING CLASSES FROM ABOVE ARE GIVEN ON THE FOLLOWING PAGE**
# TIMES AND INSTRUCTORS OF TOP TEN CLASSES

**CMPT 181 B**  
INTRODUCTION TO DATA STRUCTURES AND NONNUMERIC COMPUTING  
*AM* TERM SEC 01 M.W.F. 9:15U-10:30 COMM 144 J. TREMULAY  
*BM* TERM SEC 02 M.W.F. 10:30-11:45 ARTS 256 B. HULMUND

**CMPT 212 AORB**  
INTRODUCTION TO SYSTEMS PROGRAMMING  
*AM* TERM SEC 01 M.W.F. 8:15U-9:30 COMM 143 I. VOLK  
*BM* TERM SEC 02 M.W.F. 9:45U-11:00 COMM 142 D. COLE

**CMPT 215 AORB**  
COMPUTER PROGRAMMING LANGUAGES  
*AM* TERM SEC 01 M.W.F. 3:30U-4:30 ARTS 258 K. BOOTH

**CMPT 378 A**  
SIMULATION PRINCIPLES  
*AM* TERM SEC 01 M.T.W. 2:15-3:30 COMM 245 J. COOKE  
SEC 02 Th. 7:15U-8:30WPR COMM 45 C. SHANKS

**MATH 311 A**  
NUMERICAL ANALYSIS  
SEE MATH DEPARTMENT FOR TIMES OF THIS CLASS

**CMPT 479 B**  
SIMULATION APPLICATIONS II  
*BM* TERM SEC 01 Th. 7:15U-8:30WPR COMM 44 C. SHANKS

**CMPT 478 B**  
REAL-TIME INFORMATION SYSTEMS  
*BM* TERM SEC 01 M.W. 11:30U-12:15U COMM 112 G. HEFER

**CMPT 380 B**  
DECISION MODELS FOR COMPUTER SYSTEM MANAGEMENT  
*BM* TERM SEC 01 M.T.W. 4:15U-5:30 COMM 145 J. COOKE
SCHEDULE 2. IF YOU ARE NOT PLANNING TO TAKE ADDITIONAL CLASSES 
AFTER THE COMING YEAR

SUGGESTED CLASSES FOR DON SCHELLENBURG 693716

<table>
<thead>
<tr>
<th>CLASS</th>
<th>FINAL RATING</th>
<th>REQUIRED CLASS</th>
<th>RECOMM OPTION BASED UPON INTEREST</th>
<th>STUDENT INTEREST RATING</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 101</td>
<td><em>1020</em></td>
<td>X</td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>CMPT 378 A</td>
<td><em>1010</em></td>
<td>X</td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>CMPT 215</td>
<td><em>1580</em></td>
<td>X</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>CMPT 212</td>
<td><em>1540</em></td>
<td>X</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>MATH 311 A</td>
<td><em>1500</em></td>
<td>X</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>NOTE 1 CMPT 479 B</td>
<td><em>130</em></td>
<td></td>
<td></td>
<td>130</td>
</tr>
<tr>
<td>CMPT 479 B</td>
<td><em>50</em></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>CMPT 380 B</td>
<td><em>30</em></td>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

NOTE 1 THE FOLLOWING A HALF CLASSES MUST BE TAKEN BEFORE THIS B HALF CLASS 
CMPT 378

TIMES AND INSTRUCTORS OF THE TOP TEN RANKING CLASSES FROM ABOVE ARE GIVEN ON THE FOLLOWING PAGE
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Section</th>
<th>Time</th>
<th>Location</th>
<th>Instructor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 181 B</td>
<td>Introduction to Data Structures and Numerical Computing</td>
<td>A</td>
<td>MWF, 9:30-10:15</td>
<td>COMM 146</td>
<td>J. Tremblay</td>
</tr>
<tr>
<td>CMPT 181 B</td>
<td>Introduction to Data Structures and Numerical Computing</td>
<td>B</td>
<td>MWF, 11:30-12:15</td>
<td>ARTS 258</td>
<td>B. Holmlund</td>
</tr>
<tr>
<td>CMPT 370 A</td>
<td>Simulation Principles I</td>
<td>A</td>
<td>Tu-Th, 2:30-4:15</td>
<td>COMM 245</td>
<td>J. Coupe</td>
</tr>
<tr>
<td>CMPT 370 A</td>
<td>Simulation Principles I</td>
<td>B</td>
<td>Tu-Th, 12:30-2:15</td>
<td>COMM 245</td>
<td>G. Snickers</td>
</tr>
<tr>
<td>CMPT 213 A/M</td>
<td>Computer Programming Languages</td>
<td>A</td>
<td>MWF, 3:30-4:15</td>
<td>ARTS 258</td>
<td>K. Booth</td>
</tr>
<tr>
<td>CMPT 212 A/M</td>
<td>Introduction to Systems Programming</td>
<td>A</td>
<td>MWF, 9:15-10:00</td>
<td>COMM 142</td>
<td>R. Vuk</td>
</tr>
<tr>
<td>MATH 311 A</td>
<td>Numerical Analysis</td>
<td></td>
<td></td>
<td></td>
<td>Math Department</td>
</tr>
<tr>
<td>CMPT 479 B</td>
<td>Simulation Applications II</td>
<td>A</td>
<td>Tu-Th, 11:00-12:00</td>
<td>COMM 44</td>
<td>G. Shanks</td>
</tr>
<tr>
<td>CMPT 476 B</td>
<td>Real-Time Information Systems</td>
<td>A</td>
<td>MWF, 11:15-12:00</td>
<td>COMM 142</td>
<td>G. Heater</td>
</tr>
<tr>
<td>CMPT 380 B</td>
<td>Decision Models for Computer System Management</td>
<td>A</td>
<td>Tu-Th, 4:00-5:15</td>
<td>COMM 145</td>
<td>J. Coupe</td>
</tr>
</tbody>
</table>
The first page of ADVISOR's output is similar to that of the non-major with the exception of the indication that the student is majoring in this field.

The second page was not present in the first example. This page is a listing of the required classes for the major and is therefore not printed for non-major.

The third page gives the classes taken as submitted by the student as well as a listing of those classes that the student had received permission to take.

The fourth page indicates the student's interest ratings—note that the revised format of five ratings is in effect for this example.

The instruction page for the major is the same as that for the non-major.

Schedule 1, however, has been expanded considerably for the major student. Four additional columns have been added with the following headings:

- REQUIRED CLASS
- PREREQ FOR A REQ'D CLASS
- REQ'D OPTION BASED UPON INTEREST
- PREREQ FOR A REQ'D OPTION

An "x" under one of the headings indicates that a heavy weight (1500) has been added to that class's total rating. In the example, CMPT 181B has a total rating of 3260 comprised of: 1500 for being a required class; 1500 for being a prerequisite for another required
class; 120 for interest rating; and the remaining 140 for being a prerequisite for a senior class with an above-average interest rating.

In examining Schedule 1, it is apparent that this student has a total of seven computer science classes for which he has either satisfied the prerequisites or been given special permission. The schedule indicates to the student that four of these classes are required classes, and one other is contained in the list of required classes. In analyzing this example, the student would probably be wise to select at least the first five classes in the list.

In Schedule 2, as with the non-major, considerations for satisfying prerequisites of senior classes are ignored. As a result, the interest weighting has a more significant effect on the ranking of the classes.

4.7 Testing the Pilot System

4.7.1 The Test Plan

The second phase of the pilot study was the testing of the system using real students. The original plan was to use students enrolled in summer sessions as the subjects of the test. However, due to programming delays, the test plan was altered to utilize students attending the fall term.
The objectives of the test were twofold:
1. to evaluate the system under real conditions
2. to ascertain the student's response to such a system.

The test was conducted in September of 1971 using the students enrolled in three different computer science classes: CMPT 212, CMPT 378 and CMPT 417. The students were asked to "wind back the clock" and pretend that they were using the system to register for classes for the present term.

At the conclusion of a brief introduction to the nature of the experiment, forms for both INFORMER and ADVISOR were distributed to each student, filled out, and then collected.

The completed forms were subsequently keypunched and submitted in separate batches to INFORMER and ADVISOR. The output was separated by student and returned to the students at the next meeting of the class. The students were given some time to analyze their output and then were asked to answer a questionnaire on the usefulness of the system to them. The questionnaires and all system output were then collected for analysis.

The students enrolled in the three classes represented a reasonable cross-section of students taking classes from the Department of Computational Science. In total, there were 63 students comprised of the following groups:

<table>
<thead>
<tr>
<th>College</th>
<th>Major</th>
<th>Non-major</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commerce</td>
<td>14</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>Arts and Science</td>
<td>34</td>
<td>9</td>
<td>43</td>
</tr>
<tr>
<td>Other</td>
<td>48</td>
<td>15</td>
<td>63</td>
</tr>
</tbody>
</table>
Approximately 75% of the students were in their second or third years at University.

The following questions were asked of the students after they had completed the input forms:

1. How long did it take to fill out the forms?
2. How long did it take you to decide on classes last year?
   - less than 5 hours
   - 5-10 hours
   - more than 10 hours
3. Did you have difficulty understanding how to use this system?
4. Were there any interest topics that you did not understand? Which ones?

When the output from both ADVISOR and INFORMER was returned to the students, the students were asked to indicate their answers to the following questions:

1. Did you have any difficulty in understanding ADVISOR's instructions?
2. Do you have any suggestions for improvements?
3. Would you use this system if it were available?
   - ADVISOR?
   - INFORMER?
4. Do you feel this system would save you any time when selecting classes?
5. How close did the system come to suggesting the classes you are now taking?
6. Do you know of any reason why the system did not match your selections?

7. Was the class information provided by INFORMER
    Insufficient
    Enough
    Too much

The students were also asked to give a list of all classes that they were taking in the coming year. This information would allow the comparison of the suggested classes with what the student actually decided to take, prior to the use of the system.

4.7.2 Problems Encountered

The tests were conducted rather smoothly with only minor problems. An example of one of the problems that did occur is that for the first batch of students a program "bug" caused an incorrect college to be printed on the first page. The problem was corrected for the subsequent runs.

One adjustment that was made "mid-stream" was the addition of the "hate" interest rating for topics. This was a result of some comments from students after the first class was tested. The fact that some students were given special permission to take classes for which they had not satisfied the prerequisites was also overlooked and subsequently discovered during the test. Coding was added to ADVISOR to handle such instances. The program was changed to accept input identifying a class
which the student has been allowed to take and to immediately give
that class potential status.

It was also discovered during the test phase that there were
some errors in the data contained in the files. One example was
CMPT 313 being set up as an "A" half class only, while in actual fact
a "B" half class was also offered. The error was traced back to the
source document from which the class list file was built. That document
showed the class as being an "A" half class only. As a result, the
class was not suggested in cases where it should have been. It should
be mentioned that in an actual production system, an error of this type
should have been caught by the pre-screening done by the class instructors.
CHAPTER V

EVALUATION OF THE PILOT SYSTEM

5.1 Student Feedback

As mentioned in the previous chapter, the test students were presented with two questionnaires. The first questionnaire was designed to ascertain the student's reaction to the input forms as well as gain an indication of the student's concept of the length of time spent in previous years on the task of course selection. The second set of questions dealt directly with the usefulness of the system to the student.

The summarized results of the students' answers to the first questionnaire are presented in Table 5.1. A total of 53 students answered the questionnaire.

It can be seen from the results of this questionnaire that the majority (85%) of the students did not have trouble understanding the use of the forms. However, 20% of the students had problems interpreting at least one of the interest topics.

The majority of the students took between 10-20 minutes to complete the forms for both ADVISOR and INFORMER. The survey revealed that approximately 42% of the students had spent more than five hours on course selection in the previous year.

The summarized results of the second questionnaire are as follows:
### TABLE 5.1
RESPONSES TO FIRST QUESTIONNAIRE

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long did it take you to fill out the forms?</td>
<td>CMPT 212</td>
</tr>
<tr>
<td>less than 5 minutes</td>
<td>CMPT 378</td>
</tr>
<tr>
<td>5-10 minutes</td>
<td>CMPT 417</td>
</tr>
<tr>
<td>10-20 minutes</td>
<td>TOTAL 10</td>
</tr>
<tr>
<td>How long did it take you to decide on classes last year?</td>
<td>CMPT 142</td>
</tr>
<tr>
<td>less than 5 hours</td>
<td>CMPT 126</td>
</tr>
<tr>
<td>5-10 hours</td>
<td>CMPT 35</td>
</tr>
<tr>
<td>over 10 hours</td>
<td>CMPT 24</td>
</tr>
<tr>
<td>Did you have difficulty understanding how to use the system?</td>
<td>CMPT 4</td>
</tr>
<tr>
<td>YES</td>
<td>CMPT 3</td>
</tr>
<tr>
<td>NO</td>
<td>CMPT 23</td>
</tr>
<tr>
<td>Were there any interest topics you did not understand?</td>
<td>CMPT 5</td>
</tr>
<tr>
<td>YES</td>
<td>CMPT 6</td>
</tr>
<tr>
<td>NO</td>
<td>CMPT 22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPT 4</td>
<td>CMPT 23</td>
</tr>
</tbody>
</table>
1. ADVISOR's instructions adequate? YES - 84% NO - 16%

2. Any suggestions for improvement? YES - 40% NO - 60%

3. Would you use ADVISOR? YES - 95% NO - 5%
   INFORMER? YES - 95% NO - 5%

4. Would the system save time in selecting classes? YES - 90% NO - 10%

5. How close did ADVISOR come to picking the same classes? NO - 8% RIGHT - 4% CLOSE - 64% FAIR - 13% NOT CLOSE - 13%

6. Do you know of any special reason for the difference? YES - 23% NO - 67%

7. Did INFORMER provide enough information? YES - 70% NO - 20% NO ANS. - 10%

In analyzing the results of the second questionnaire, a number of points should be mentioned. The one point that stands out the most is the acceptance of such a system as being worthwhile by the students. This fact is clearly made by the response to questions 3 and 4. Ninety-five percent (57 students) said they would use ADVISOR. An identical number said they would use INFORMER. Only one student said he would not use either ADVISOR or INFORMER if they were available. Question 4 reveals that 90% felt that the system would save them time when deciding on classes. In comparing the results of the two questions, it is interesting to note that 5% of the students would use the system in spite of the fact that they thought there wouldn't be any time saving.
As can be seen from the results of question 2, a large number of the students offered suggestions for improvements, which certainly indicates an interest in the concept.

In analyzing the responses to questions 5 and 6 regarding the classes suggested by ADVISOR, it is obvious that the students thought that ADVISOR did a relatively good job. Question 6 resulted in some very interesting comments. One alarming example is that ten of the students felt the reason for the difference between what they had registered in and what ADVISOR had suggested was due totally to errors on their part when registering without the help of ADVISOR. For example, some of these students indicated that they had not understood what a particular class covered when they had enrolled in it. Some students also indicated that they had received special permission to take a class concurrently with its prerequisites, a fact that ADVISOR had not been made aware of. Another reason that was given for differences was the failure on the part of the student to give an accurate rating of his interests in the topics.

In summary, the results of the second questionnaire indicate without a doubt the students' evaluation of the pilot system as being very worthwhile and useful to them when selecting classes.

5.2 Analysis of ADVISOR's Suggestions

It was mentioned in the discussion of the test plan that the students were asked to "pretend" that they were registering for the
coming term. They were also asked to list the classes in which they had actually been registered. The purpose behind both of these actions was to allow a comparison of the set of classes suggested by ADVISOR with those selected by the student through the conventional course selection procedures.

A comparison of ADVISOR's suggested set of classes with those submitted by each student was performed using the following guidelines:

1. Schedule 1 was used, assuming all students would be taking classes from this department in subsequent years.

2. The number of relevant Computer Science and Math classes determined the number to be selected from the top of the list of ADVISOR's suggested classes.

3. The scoring was on a percentage basis. For example, if the student registered in six relevant classes, three of which were in the top six suggested by ADVISOR, then the scoring would be 50%. If ADVISOR only suggested one out of the six, then the scoring would be 17%.

The distribution of the results of the comparison of ADVISOR's list with those classes chosen by the student are presented in Table 5.2. The table indicates, for example, that for 19 students in the CMPT 212 test class ADVISOR suggested between 50-75% of the classes chosen by the student. It should be re-emphasized that the number of classes examined at the top of ADVISOR's list was determined by the number of relevant classes chosen by the student.
<table>
<thead>
<tr>
<th>ADVISOR's Suggestions</th>
<th>CMPT 212</th>
<th>CMPT 378</th>
<th>CMPT 417</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1-24%</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>25-49%</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>50-74%</td>
<td>19</td>
<td>9</td>
<td>3</td>
<td>31</td>
</tr>
<tr>
<td>75-99%</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>100%</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

**TABLE 5.2**

**COMPARISON OF ADVISOR'S SUGGESTIONS WITH STUDENTS' CHOICES**
It should also be noted that seven of the students were not examined because of incomplete information. The results in Table 5.2 have not been adjusted for the errors in the data or for classes that were taken without their prerequisites being satisfied.

Another synopsis of the results of the analysis is presented in Table 5.3. The classes chosen by the students are totalled for each class, as well as the number of matches with ADVISOR's list. For example, for the CMPT 212 test class ADVISOR suggested 85 out of 143 classes or 59%. The table also indicates the number of classes taken without their prerequisites being satisfied. For example, in the CMPT 212 test class there were 15 such instances that ADVISOR was not informed about. As well, the number of instances where students had enrolled in CMPT 313B were noted. It should be recalled that CMPT 313 had been set up in the class list file as an "A" half class only and therefore ADVISOR missed suggesting it. The table indicates the adjusted percentages in each test class and for the total sample as well.

5.3 System Evaluation

The analysis of the second questionnaire revealed that the students were quite impressed with the pilot system. The intent of this section is to discuss the system from the technical point of view.

In general terms, the pilot system was quite effective. ADVISOR's suggested set of classes was usually quite acceptable and
### TABLE 5.3

**SUMMARY OF ADVISOR'S EFFECTIVENESS**

<table>
<thead>
<tr>
<th></th>
<th>CMPT 212</th>
<th>CMPT 378</th>
<th>CMPT 417</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of classes taken</td>
<td>143</td>
<td>66</td>
<td>44</td>
<td>253</td>
</tr>
<tr>
<td>Total # of matches</td>
<td>85</td>
<td>37</td>
<td>19</td>
<td>141</td>
</tr>
<tr>
<td>%</td>
<td>59%</td>
<td>56%</td>
<td>48%</td>
<td>56%</td>
</tr>
<tr>
<td>Special permission classes</td>
<td>15</td>
<td>4</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>Total # of CMPT 313 missed</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>Revised total matches</td>
<td>113</td>
<td>42</td>
<td>28</td>
<td>183</td>
</tr>
<tr>
<td>Revised %</td>
<td>78%</td>
<td>63%</td>
<td>64%</td>
<td>68%</td>
</tr>
</tbody>
</table>
ensured that the student satisfied the requirements for the major. In light of the errors in the data, the limited experience of the individual who set up the interrelationships of similar classes and the questionable choice of interest topics, the pilot system worked out well. However, given strengths in place of these weaknesses, ADVISOR could do a much better job and should be able to suggest a faultless set of classes for each student.

In specific terms, the set of programs that make up ADVISOR are far from perfect. FORTRAN is not the best language to use for such a system, mainly because of editing functions that can be performed more effectively by an input-output oriented language such as COBOL. Again, because FORTRAN was used, the input formats for the files are perhaps not the most efficient from a data entry standpoint. However, for the purposes of the pilot study, FORTRAN offered the ease of coding the handling of rather complex three dimensional arrays, indirect subscripting, etc., that can be found in the program listings. Perhaps, as an afterthought, the best combination would be to use COBOL for editing and report purposes, and to use FORTRAN for the "number crunching."

In terms of execution time, ADVISOR was quite efficient taking approximately 25 seconds to process each student (in 1971). In today's environment that time should be reduced significantly.
5.4 Evaluation of the Batch Approach

The intent of this section is to discuss the ramifications of utilizing the batch approach for an expanded course-advising system for many departments of the university.

It was pointed out previously that the reason that the batch approach was used for the pilot study was that the resources required to develop an on-line system were considerably greater than the batch. As it turned out, the pilot system appeared to be quite acceptable to the students. However, the study did not give any indication of whether the students would favour an on-line system over the batch.

The main advantage of the batch system is its low cost. The cost is lower for batch systems in a number of key areas: program development, execution time and additional equipment requirements. Program development costs are lower because the batch programs are more simple in nature. Execution time is less because running all the requests for one department requires less file accesses. Additional equipment requirements are non-existent for the batch approach.

The main drawback of the batch approach is the function of input preparation and the distribution of output. It is impractical to ask the students to learn how to keypunch their requests. Therefore, a data entry function must be provided.

In the pilot study, the data entry function was performed by collecting the completed forms from the students, taking the forms to the computer center's keypunch section, keypunching the requests onto
cards, collecting all of punched cards and submitting them to the system, receiving the output, separating the output for each student and finally distributing the output to the students. In the expanded system, this type of central collection and distribution could be centered around the individual college offices, for example. The students would drop off their completed forms and pick up their output the next day.

An alternative to the central collection and data entry scheme would be distributed data entry facilities. This concept would entail data entry personnel to be assigned to various locations on campus to keypunch student input. The punched cards could be given to the student to submit to another central collection depot for subsequent processing in batches.

In summary, the batch approach was ideal for the pilot study because it was relatively inexpensive to develop and operate. The batch approach could be used in a production environment if suitable data entry facilities could be arranged.

5.5 Achieving the Ideal System

The concept of the ideal course-advising and information system was developed in the Introduction. In reviewing the list of ten criteria, it is evident that the pilot system falls short of being the ideal system by virtue of not meeting the following:

1) be available at a number of locations on campus,
2) serve a student in any college on campus,
3) retrieve information about the student's past (from a source other than the student),
4) suggest a timetable of classes for the student.

Addressing the second criteria first, it should be re-emphasized that the function of the pilot system was to be the basis for the pilot study which was concerned with course advising for only one department. However, with minor changes, the pilot system could operate for any number of colleges under the batch processing concept. The same programs would be used for all colleges, but the contents of the files would have to include data for the classes of each department. The resulting problem is to determine the most efficient method of organizing the contents of the files. There are two main alternative methods. One alternative would be to have separate set of files for each department. In this way, each department would be independently responsible for the data contained in the complete set of files used for advising its students, and would not have to rely on other departments to update their classes. However, the problems caused by such a scheme far outweigh this advantage. The number of different files required would be enormous. As well, one class may be relevant to a number of different departments and, as a result, would appear in a number of different sets of files. Also, the records containing class information from another department would be maintained by individuals who may not be familiar with the contents of the class. In addition, under the
batch concept, separate program runs for each department would be necessitated utilizing appropriate JCL to identify a specific department's files.

The better alternative is to have one large data base for each file containing the classes from all departments. As a result, a class would only appear once in each file. Also, one set of JCL could be used allowing all departments to be processed in one run under the batch concept. Each class could have a designated department responsible for its maintenance and security codes could be set up to allow only the designated departments to change the contents of a class record. In this way, the integrity of the data could be protected.

The criteria that the system be available at a number of locations on campus could be met using the batch concept if collection depots were established in the various college offices.

The third criteria missing from the pilot system was the retrieval of the student's record from a source other than the student. Presumably this would entail the establishment of a student record file that was available to ADVISOR. The program could be easily modified to access this file and retrieve the successfully completed classes.

The concept of scheduling a timetable for the student was included as one criterion of the ideal system because of the length of time spent on this function by the student. It is usually a complex task, and if it were to be performed by the system, a method of identifying all classes to be taken by the student would have to be developed.
The pilot system is designed to operate on the departmental level, and does not take into consideration unrelated classes taken from other departments. As a result, the system would have to be informed in some manner of the other classes the student has already decided to take. One alternative would be to have a separate scheduling program that would accept as input the total list of classes finally decided upon by the student. The program would access a class time file and attempt to minimize conflicts in the student's timetable. The system would leave to the student the problem of resolving conflicts that would require class changes.
Chapter VI

IMPLICATIONS OF THE STUDY

6.1 Conclusions of the Pilot Study

The previous chapter reviewed the results of the pilot study and provided an evaluation of the pilot system from various viewpoints. The intent of this final chapter is to draw conclusions from the pilot study and to discuss their implications. The purpose of this particular section is to highlight the findings of the pilot study.

There are three main conclusions that can be drawn from the pilot study:

1. there was a strong indication from the students that the need exists for an improved course-advising system
2. the problem of routine course selection can be satisfactorily handled by a computer-based system
3. that if such a system were available it would receive a high degree of acceptance and usage by the students.

The first conclusion is drawn mainly from the fact that 90% of the students felt that such a system would save them time in selecting classes. It is also re-enforced by the fact that 95% of the students said they would use the system if it were available.

The conclusion that a computer-based system could be used for course selection was demonstrated by the pilot study. This conclusion should perhaps be qualified by stating that the effectiveness of the
system is definitely dependent upon the correctness of the data in the system files and the amount of effort placed in arriving at an adequately defined set of interest topics. However, the pilot study did in fact demonstrate that the functions involved in course selection such as considering prerequisites, required classes and the student's interests could be performed by a computer-based system.

The third conclusion of the pilot study that the system would receive a high degree of usage and acceptance by the students was pointed out very clearly by the fact that 95% of the students in the study said that they would use the system if it were available. This factor should not be overlooked because without the support of the students, an attempt to provide such a system on campus-wide scale would be a waste of time.

In summary, the pilot study has demonstrated:

a) a need exists for such a system
b) that the system will work
c) the system will be used and accepted by the students.

6.2 Limitations of the Pilot Study

The previous section discussed the conclusions that can be derived from the pilot study. The purpose of this section is to discuss certain limitations of the pilot study that should be taken into consideration when analyzing the implications of the study.

One important limitation of the pilot study is the fact that the study was performed for only one of many departments at the campus.
The reasons for choosing the Department of Computational Science were discussed in the Introduction. To briefly review, this department contained the vital characteristics that such a system would have to be able to deal with: a prerequisite structure, required classes for major and students enrolled from a number of colleges. It is possible that because of unforeseen differences between departments that such a system might have been developed differently if another department had been chosen for the pilot study.

Another important limitation of the pilot study is the fact that it was performed in 1971. During the five years that have elapsed between the conducting of the pilot study and the writing of this thesis, there may have been improvements made to the course selection process such that student responses might be different if the pilot study had been done more recently.

It should also be re-emphasized that the function of the pilot system was limited to helping students with the problem of routine course selection only. In other words, it is assumed that through counselling or some other means that the student knows what college he should be in and what major he desires. The pilot study did not attempt to evaluate computer-assisted counselling systems.

Another factor that should be mentioned is that the students involved in the pilot study were all enrolled in at least one computer science class and therefore could conceivably have been more receptive and interested in a computer-based advising system than other students. Therefore, it is possible that the responses by the students were somewhat biased in favour of the system.
Lastly, the pilot study did not attempt to determine the costs of developing and maintaining such a system. The costs of such a system are totally dependent upon the system approach that is taken which in turn is dependent upon short and long term computing objectives of the University.

6.3 Implications of the Pilot Study

The previous two sections of this chapter have discussed the conclusions of the pilot study as well as some cautions to be taken into consideration when interpreting these conclusions. The purpose of this section is to discuss the implications of the findings of the pilot study.

The major implication of the pilot study is that because it has been demonstrated that a computer-based course-advising and information system is technically feasible that further consideration should be given to a campus-wide system.

There are a number of factors that should be considered in greater depth before such a system is developed. Firstly, the present course selection procedures should be re-examined and student feedback should be requested to see if they still feel that the present system could be improved.

Secondly, the costs of a campus-wide system should be analyzed in light of other computer applications on campus. The comparison of manual versus computer-based system costs should be performed to confirm
that there is in fact an economic and/or service advantage resulting from the computer-based alternative.

Implicit in determining the costs of a total system is determining the size of the necessary data base to handle all departments and how that data base should be maintained. Additional comments on this area can be found in subsection 5.5.

In conclusion, the author of this thesis recommends that based upon the positive results of this pilot study that the concept of a computer-based course-advising and information system should not be dropped and filed away with this thesis but that someone should "pick up the ball" and run with it a little further with the hope of providing a better course-advising system for the mutual benefit of the University and its students.
BIBLIOGRAPHY


Schuessel, G. "When Not to Use a Data Base," Datamation, Nov. 1975.


$104  WATFIV  900471.R.HEWITT,TIMF=10,PAGES=75
1  INTEGER TITLE(61)
2  DIMENSION KODE(15),HRFG(10),1TOPI(12),ICOVRE(50)
3  DIMENSION NSAT(120)
4  DIMENSION SPACE(1050),LOC(1050)
5  DEFINE FILE100(100,1000,L,1D11),110(100,1000,L,1D11),
6  D12(100,1000,L,1D13),13(500,61,L,1D15)
7  DEFINE FILE 9(324,80,L,1D16)
8  READ(5,4)(KODE(11),I=1,15)
9  4  FORMAT(15I7)
10  KEY=1
11  5  READ(KEY)
12  IF(EOF,.01,STOP)
13  SN T04109,150,700,250,300,350,400,450,500,550,650),L
C
CLASS LIST FILE CREATION

14  100 DO 117 I=1,100
15  READ(6,117)I,N2,NS,NT,TITLE
16  1  FORMAT(24,12,11,61A1)
17  IF(EOF,.01,STOP)
18  117 WRITE(I,117)I,N2,NS,NT,TITLE
19  WRITE(6,21)
20  21  FORMAT(15I7,'SOMETHING WRONG-MORE THAN 100\CLASSSES')
21  STOP
C
WRITE OUT CONTENTS OF CLASS LIST FILE

22  120 REWIND 1
23  07 DO 130 I=1,100
26  130 WRITE(I,130)I,N2,NS,NT,TITLE
27  130 WRITE(I,130)I,N2,NS,NT,TITLE
28  130 WRITE(I,130)I,N2,NS,NT,TITLE
29  130 WRITE(I,130)I,N2,NS,NT,TITLE
30  140 STOP
C
PREQUALITE FILE CREATION

31  150 DO 160 I=1,100
32  READ(I,51)I,L1,L2,L3,L4
33  5  FORMAT(24,12,11,61A1)
34  IF(EOF,.01,STOP)
35  160 WRITE(I,160)I,L1,L2,L3,L4
36  WRITE(I,160)I,L1,L2,L3,L4
37  160 WRITE(I,160)I,L1,L2,L3,L4
38  160 WRITE(I,160)I,L1,L2,L3,L4
39  160 WRITE(I,160)I,L1,L2,L3,L4
40  160 WRITE(I,160)I,L1,L2,L3,L4
41  160 WRITE(I,160)I,L1,L2,L3,L4
42  160 WRITE(I,160)I,L1,L2,L3,L4
43  160 WRITE(I,160)I,L1,L2,L3,L4
44  160 WRITE(I,160)I,L1,L2,L3,L4
45  160 WRITE(I,160)I,L1,L2,L3,L4
46  160 WRITE(I,160)I,L1,L2,L3,L4
47  160 WRITE(I,160)I,L1,L2,L3,L4
48  160 WRITE(I,160)I,L1,L2,L3,L4
49  160 WRITE(I,160)I,L1,L2,L3,L4
50  160 WRITE(I,160)I,L1,L2,L3,L4
51  160 WRITE(I,160)I,L1,L2,L3,L4
52  160 WRITE(I,160)I,L1,L2,L3,L4
53  160 WRITE(I,160)I,L1,L2,L3,L4
54  160 WRITE(I,160)I,L1,L2,L3,L4
55  160 WRITE(I,160)I,L1,L2,L3,L4
56  160 WRITE(I,160)I,L1,L2,L3,L4
57  160 WRITE(I,160)I,L1,L2,L3,L4
58  160 WRITE(I,160)I,L1,L2,L3,L4
59  160 WRITE(I,160)I,L1,L2,L3,L4
60  160 WRITE(I,160)I,L1,L2,L3,L4
61  160 WRITE(I,160)I,L1,L2,L3,L4
62  160 WRITE(I,160)I,L1,L2,L3,L4
63  160 WRITE(I,160)I,L1,L2,L3,L4
64  160 WRITE(I,160)I,L1,L2,L3,L4
65  160 WRITE(I,160)I,L1,L2,L3,L4
66  160 WRITE(I,160)I,L1,L2,L3,L4
67  160 WRITE(I,160)I,L1,L2,L3,L4
68  160 WRITE(I,160)I,L1,L2,L3,L4
69  160 WRITE(I,160)I,L1,L2,L3,L4
70  160 WRITE(I,160)I,L1,L2,L3,L4
71  160 WRITE(I,160)I,L1,L2,L3,L4
72  160 WRITE(I,160)I,L1,L2,L3,L4
73  160 WRITE(I,160)I,L1,L2,L3,L4
74  160 WRITE(I,160)I,L1,L2,L3,L4
75  160 WRITE(I,160)I,L1,L2,L3,L4
76  160 WRITE(I,160)I,L1,L2,L3,L4
77  160 WRITE(I,160)I,L1,L2,L3,L4
78  160 WRITE(I,160)I,L1,L2,L3,L4
79  160 WRITE(I,160)I,L1,L2,L3,L4
80  160 WRITE(I,160)I,L1,L2,L3,L4
81  160 WRITE(I,160)I,L1,L2,L3,L4
82  160 WRITE(I,160)I,L1,L2,L3,L4
83  160 WRITE(I,160)I,L1,L2,L3,L4
84  160 WRITE(I,160)I,L1,L2,L3,L4
85  160 WRITE(I,160)I,L1,L2,L3,L4
86  160 WRITE(I,160)I,L1,L2,L3,L4
87  160 WRITE(I,160)I,L1,L2,L3,L4
88  160 WRITE(I,160)I,L1,L2,L3,L4
89  160 WRITE(I,160)I,L1,L2,L3,L4
90  160 WRITE(I,160)I,L1,L2,L3,L4
91  160 WRITE(I,160)I,L1,L2,L3,L4
92  160 WRITE(I,160)I,L1,L2,L3,L4
93  160 WRITE(I,160)I,L1,L2,L3,L4
94  160 WRITE(I,160)I,L1,L2,L3,L4
95  160 WRITE(I,160)I,L1,L2,L3,L4
96  160 WRITE(I,160)I,L1,L2,L3,L4
97  160 WRITE(I,160)I,L1,L2,L3,L4
98  160 WRITE(I,160)I,L1,L2,L3,L4
99  160 WRITE(I,160)I,L1,L2,L3,L4
100  160 WRITE(I,160)I,L1,L2,L3,L4
101  160 WRITE(I,160)I,L1,L2,L3,L4
102  160 WRITE(I,160)I,L1,L2,L3,L4
103  160 WRITE(I,160)I,L1,L2,L3,L4
104  160 WRITE(I,160)I,L1,L2,L3,L4
105  160 WRITE(I,160)I,L1,L2,L3,L4
106  160 WRITE(I,160)I,L1,L2,L3,L4
107  160 WRITE(I,160)I,L1,L2,L3,L4
108  160 WRITE(I,160)I,L1,L2,L3,L4
109  160 WRITE(I,160)I,L1,L2,L3,L4
110  160 WRITE(I,160)I,L1,L2,L3,L4
111  160 WRITE(I,160)I,L1,L2,L3,L4
112  160 WRITE(I,160)I,L1,L2,L3,L4
113  160 WRITE(I,160)I,L1,L2,L3,L4
114  160 WRITE(I,160)I,L1,L2,L3,L4
115  160 WRITE(I,160)I,L1,L2,L3,L4
116  160 WRITE(I,160)I,L1,L2,L3,L4
117  160 WRITE(I,160)I,L1,L2,L3,L4
118  160 WRITE(I,160)I,L1,L2,L3,L4
119  160 WRITE(I,160)I,L1,L2,L3,L4
120  160 WRITE(I,160)I,L1,L2,L3,L4
121  160 WRITE(I,160)I,L1,L2,L3,L4
122  160 WRITE(I,160)I,L1,L2,L3,L4

REQUIRED CLASSIFICATION CREATION

C
READ TV NO OF COLLEGES

54  297 READ5,91N
55  WRITE(1,91N)
56  91  FORMAT(17)
57  DO 230,I=1,N
READ NO. OF OPTIONS FOR THIS COLLEGE AND RECORD ON FILE

READ(5,811NO)
WRITE(3,911NO
IF(NO.EQ.110 TO 220

READ IN OPTIONS AND RECORD ON FILE

DO 210 K=1,NO.
READ(5,911NOPT,NOUTOF
WRITE(3,911NOPT,NOUTOF
9 FORMAT(12I2)

READ IN OPTION CLASSES (IF ANY) AND RECORD ON FILE

READ(5,111J(NRED(J),J=1,NOUTOF
10 FORMAT(11I2)
WRITE(3,111J(NRED(J),J=1,NOUTOF

READ IN REQUIRED CLASSES FOR THIS COLLEGE AND RECORD ON FILE

220 READ(5,171J(NRED(J),J=1,10)
WRITE(3,171J(NRED(J),J=1,10)

PRINT OUT CONTENTS OF REQUIRED CLASSES FILE

REINDE
WRITE(6,11)
11 FORMAT(*X,10X,*REQUIRED CLASSES FILE*)
READ(3,911NO
235 READ(3,911NO
WRITE(6,121J,NO
12 FORMAT(*X,10X,*NO. OF OPTIONS FOR COLLEGE* #13, * IS * 121
IF(NO.EQ.110 GOTO245
DO 240 J=1,NO.
READ(5,911NOOUT,NOUTOF
WRITE(6,13)J,NOUT,NOUTOF
13 FORMAT(*X,10X,*OPTION NO. * #11,X,*STUDENT MUST CHOOSE * 12,* OF THE FOLLOWING * 12,* CLASSES*)
READ(3,111J(NRED(K),K=1,NOUTOF
WRITE(6,14)J,K,NRED(K)
14 FORMAT(*X,20X,10(J12,3X))
245 WRITE(6,1511
15 FORMAT(*X,10X,*REQUIRED CLASSES FOR COLLEGE NO. * 12,* ARE:*)
READ(3,111J(NRED(K),K=1,10)
WRITE(6,16)J,K,NRED(K)
16 IF(J.NE.1 GOTO235
80 KEY=KEY+1
81 GOTO50

INTEREST TOPIC FILE CREATION OR UPDATE

READ IN NUMBER OF TOPICS

READ(1,10)
WRITE(8,91N
DO 260 T=1,N
READ(5,4)J(TPTC(J),J=1,12)
WRITE(8,4)J(TPTC(J),J=1,12)
16 FORMAT(12I4)
260 WRITE(8,1611(TPTC(J),J=1,12)

PRINT OUT TOPIC FILE
**C**

REWIND 8

WRITE(6,17)

17 FORMAT(14,1X,*INTEREST TOPIC FILE*)

READ 8,AN

WRITE(6,19IN)

18 FORMAT(14,1X,*NUMBER OF TOPICS=",",N)

WRITE(6,19)

10 FORMAT(14,1X,*THOSE TOPICS ARE:"

DO20 T=1,N

READ A,151(TOPIC(J),J=1,12)

270 WRITE(6,211(TOPIC(J),J=1,12)

27 FORMAT(*T*,7X,1244)

100 KEY=KEY+1

GO TO 50

**C**

INTEREST TOPIC COVERAGE (RELEVANCE) FILE

**C**

NOTE: CLASSFS FROM OTHER DEPTS SHOULD BE GIVEN ZERO COVERAGE

**C**

FIRST CARD INDICATES NO OF CLASSES

**C**

SECOND CARD INDICATES RATING FOR EACH TOPIC FOR FIRST CLASS

**C**

102 READ5,8INO

103 DO 329 T=1,N

104 READ5,211(ICOVGR(J),J=1,N)

105 FORMAT(591)

106 329 WRITE(4,211(ICOVGR(J),J=1,N)

**C**

PRINT OUT CONTENTS OF TOPIC COVERAGE FILE

**C**

REWIND6

WRITE(4,22)

22 FORMAT(*T*,14X,*INTEREST TOPIC COVERAGE FILE*)

WRITE(4,23)

23 FORMAT(14,1X,"CLASS NO",5X,*COVERAGE OF EACH TOPIC")

DO 339 T=1,N

339 WRITE(4,211(ICOVGR(J),J=1,N)

114 24 FORMAT(*T*,13X,12,RX,50111,1X)

115 KEY=KEY+1

116 GO TO 50

**C**

CLASS DESCRIPTIONS FILE CREATION OR UPDATE

**C**

NOTE: EACH CLASS MUST BE SEPARATED BY A BLANK CARD

**C**

WITH A $ IN COLUMN 85

**C**

FIRST CARD INDICATES NO OF CLASSES

118 355 KUP=1

119 KFILE=10

355 READ(5,81N

120 10=0

122 DO 370 L=1,4

123 370 365 I=1,1000,40

124 K=I+79

125 READ5,2511(ISPACE(J),J=1,K)

126 25 FORMAT(40A1)

127 IF(ISPACE(K),CO.,NEND)GOTO365

128 CONTINUE

129 WRITE(6,26)

130 26 FORMAT(*T*,14X,*ERROR---MORE THAN 1000 CHARACTERS OR "$" MISSING")

STOP

132 365 K=K-85

133 10=10+1
134 370 WRITE(KFILE,TD,*?71) (SPACE(I),J=1,K)
136 C PRINT OUT CONTENTS OF APPROPRIATE FILE
137 375 WRITE(6,28)
138 GOTO390
139 378 WRITE(6,37)
140 28 FORMAT(11*,20X,'CLASS DESCRIPTIONS FILE*)
141 GOTO 390
142 380 WRITE(6,29)
143 29 FORMAT(11*,20X,'CLASS ORGANIZATION FILE*)
144 GOTO390
145 385 WRITE(6,30)
146 30 FORMAT(11*,20X,'MAJOR TOPICS FILE*)
147 32 FORMAT(11*,20X,'CLASS OBJECTIVES FILE*)
148 390 DO 395 I=1,N
149 READ(KFILE,1,271)SPACE
150 395 WRITE(6,31)SPACE
151 31 FORMAT(D*,80A1)
152 KEY=KEY+1
153 GOTO50
154 400 KUP=2
155 405 KFILE=11
156 GOTO355
157 450 KUP=3
158 455 KFILE=12
159 GOTO355
160 470 KUP=4
161 475 KFILE=12
162 GOTO 355
C C TIME FILE CREATION OR UPDATE
C NOTE: TWO FILES ARE CREATED
C TIMEIDX---INDICATES LOCATION OF CLASS IN THE TIME
C FILE (SEQUENTIAL)
C TIMEFILE---CONTAINS INFO ONE SECTION PER RECORD
C (DIRECT ACCESS)
C FIRST CARD CONTAINS NO OF CLASSES
C 163 550 READ(5,81)
164 I=1
165 DO 560 I=1,N
166 WRITE(14,9)J
167 560 READ(5,33)TITLE
168 33 FORMAT(61A1)
169 WRITE(15*,J,23)TITLE
170 J=I+1
171 ITITLE(61),NFMNTI GOTO560
172 580 CONTINUE
173 J=J+1
C C PRINT OUT CONTENTS OF TIMEIDX AND TIME FILE.
C 174 WRITE(14,91)
175 RNIND14
176 WRITE(6,34)
177 34 FORMAT(11*,20X,'TIME INDEX FILE*)
178 WRITE(6,35)
179 DO 350 I=1,N
180 READ(14,91)
181 600 WRITE(6,161)J
182 15 FORMAT(I1,20X,'CLASS NO.*,5X,RECORD LOCATION')
183 36 FORMAT(D*,23X,12,14X,16)
WRITE(6,37)
185  37 FORMAT(14,2X,'TIME FILE')
185  WRITE(6,39)
187  38 FORMAT(14,2X,'CLASS NO.',5X,'CONTENTS OF FILE')
188  REMIND14
189  I=1
190  DO 660 I=1,L
191  660 READ(14,8)LOC(I)
192  DO 610 I=1,N
193  L=LOC(I)
194  K=LOC(I+1)-1
195  DO 610 J=L,K
196  READ(15,J,33)TITLE
197  610 WRITE(6,39)I,TITLE
198  39 FORMAT(14,2X,'CLASS AT 9:00 MONDAY AND SO ON')
C
C           INVERTED TIME FILE CREATION
C           FIRST RECORD CONTAINS CLASSES AT 8:30 MONDAY
C           SECOND RECORD CONTAINS CLASSES AT 9:00 MONDAY AND SO ON
C           ONE RECORD PER CARD CONTAINING ALL CLASSES AT THAT TIME
199  650 DO 670 I=1,324
200  670 READ(5,40)INSLOT(J),J=1,20)
C
C           PRINT OUT CONTENTS OF THE INVERTED TIME FILE
C
C
C
203  WRITE(6,41)
204  41 FORMAT(14,2X,'INVERTED TIME FILE')
205  WRITE(6,42)
206  42 FORMAT(14,2X,'INDIVIDUAL RECORDS FOR A TERM')
207  DO 660 I=1,L
208  READ(6,44)INSLOT(J),J=1,20)
209  660 WRITE(6,43)INSLOT(J),J=1,20)
210  43 FORMAT(14,10X,'12,7X')
211  WRITE(6,44)
212  44 FORMAT(14,2X,'INDIVIDUAL RECORDS FOR B TERM')
213  DO 690 I=163,234
214  690 READ(9,44)INSLOT(J),J=1,22)
215  660 WRITE(6,43)INSLOT(J),J=1,20)
216  43 FORMAT(14,10X,'12,7X')
217  END
APPENDIX B

INFORMER

Source Program Listing
$JOB MATFIV 900471.R.HEWETT,TIME=15,PAGES=1000

INFORMER

AUTHOR: R.H. HEWETT
DATE: SEPT. 1971

PURPOSE: A SEPARATE PROGRAM USED BY A UNIVERSITY DEPARTMENT AS AN INFORMATION SYSTEM FOR THEIR STUDENTS.

DESCRIPTION: SUPPLIES INFO OF TWO TYPES:
A: SPECIFIC INFO ON CLASSES
B: CLASSES OFFERED AT A CERTAIN TIME

SPECIFIC INFO ON CLASSES:
1. DESCRIPTION OF THE CLASS,
2. OBJECTIVES OF THE CLASS,
3. ORGANIZATION OF THE CLASS,
4. MAJOR TOPICS OF THE CLASS,
5. SECTIONS AND TIMES OFFERED.

DIMENSION HCLASS(100),INOP(5),ISPACE(1000),ITIME(160),ITIDX(100)
DIMENSION I(200),NAME(5)
DEFINE FILE 9(3240,NO,L,1016)
DEFINE FILE 11(100,1000,L,1011),12(1000,1000,L,1)
$0121,13(100,1000,L,1013),15(500,610,L,1015)
DATA MCAY,MTIME,MTERM/'MCN','TUES','WED','THUR','FRI','SUN'
'$TIHS','MON','TUE','WED','THU','FRI','SAT'
$'1130','1200','1230','1300','1330','1400','1530'
$'0900','0930','1000','1030','1100','1130','1200'
$'0430','0500','0530','0600','0630','0700','0730'
DATA YES,'MCN','PEK/YES','NO','0/'
DATA M5EMI///
DATA HANK///

READ IN TIME FILE INDEX FROM DISC
DO 90 I=1,100
90 READ(1,*,END=95)ITIDX(I)

READ IN CLASS LIST FILE FROM DISC
95 DO 100 I=1,100
100 CONTINUE
105 I=1-1

READ IN STUDENT NO AND NAME
75 READ(5,16)INST,(NAME(J),J=1,5)
16 FORMAT(10,5A4)

CHECK IF BLANK CARD (END OF STUDENTS)
C DETERMINE IF THIS INFO IS WANTED
C
26 110 READ(5,1)NWANT
27 1 FORMAT(43)
28 IF(NWANT.EQ.NNDGO TO 200
C
C READ IN NO OF CLASSES
C
29 120 READ(5,2)NCL
30 2 FORMAT(12)
C
C MAXIMUM NO OF CLASSES IS 10
C
31 IF(NCL.GT.10)NCL=10
C
C READ CLASS AND INFL. OPTIONS DESIRED
C
32 DU 170 N=1,NCL
33 READ(5,3)IC1,IC2,INOP(I,J=1,5
34 3 FORMAT(2A4,5I1)
35 WRITE(6,2000)IC1,IC2
36 2000 FORMAT('4*46(***'))
37 IF(IC1.EQ.NCLASS(1,1)AND,IC2,EQ,NCLASS(1,2))GO TO 125
38 125 CONTINUE
39 WRITE(6,5)IC1,IC2
40 5 FORMAT('S*5X,2A4,** WAS NOT FOUND IN THIS DEPARTMENT'S CLASS LIST* S')
41 GO TO 170
42 125 DU 16J K=1,5
43 IF(INOP(K),EQ.1)GO TO 170
44 IF(INOP(K),GT.5)GO TO 170
45 L=INOP(K)
46 GO TO (130,135,140,145,150),L
C
C RETRIEVE RECORD FROM CLASS DESCRIPTIONS FILE FOR CLASS NO J
C
47 130 READ(10,J,6)SPACE
48 6 FORMAT(25U1,25U1,25U1,25U1)
49 7 FORMAT(1*)
50 WRITE(6,8)SPACE
51 8 FORMAT('5*20X,5U1)
53 GO TO 160
C
C RETRIEVE RECORD FROM CLASS OBJECTIVES FILE
C
54 135 READ(13,J,6)SPACE
55 WRITE(6,8)SPACE
56 GO TO 160
C
C RETRIEVE RECORD FROM CLASS ORGANIZATIONS FILE
57  140 READ(11,J,K)ISPACE
58  WRITE(0,1)ISPACE
59  GO TO 160
C
C RETRIEVE RECORD FROM CLASS MAJORS TOPICS FILE
60  165 READ(12,J,K)ISPACE
61  LAST=1
62  DJ 148 L=1,100
63  IF(ISPACE(L).NE.200.AND.ISPACE(L).NE.100)GO TO 148
64  DJ 146 KK=LAST+1
65  IF(ISPACE(KK).NE.100)GO TO 147
66  146 CONTINUE
67  GO TO 160
68  147 WRITE(0,11)ISPACE(M),M=KK,L
69  11 FORMAT(400X,6X,20X,6UA1)
70  IF(ISPACE(L).EQ.100)GO TO 100
71  LAST=L+1
72  148 CONTINUE
73  GO TO 160
C
C TIME FILE
C
NOTE: LOCATE RECORD IN INDEX, THEN READ RECORD
74  150 LREC=ITIDX(J)
75  LNEXT=ITIDX(J+1)-1
76  WRITE(1,LW,LWCLASS(J,11),CLASS(J,2))
77  1010 FORMAT(*,G,J,SECTIONS AND INSTRUCTORS FOR *,2A4)
78  GO 155 L=LREC,LNEXT
79  READ(15*L,911TIME)
80  911 FORMAT(6UA1)
81  15 L=L,LW,LCLASS(J,11),CLASS(J,2))
82  10 L=1,L=101TIME
83  100 CONTINUE
84  170 CONTINUE
C
C PART B:
C
C CLASSES OFFERED AT A CERTAIN TIME
C
C READ IN NUMBER OF REQUESTS
C READ IN DAY, TIME, AND TERM CODES
C
85  14 FORMAT(*,20X,CLASSES OFFERED ON *,A4, AT *,A4, DURING *,A4, TIME, TERM*)
86  200 READ(51)INWANT
87  7 IF(INWANT.NE.33)GO TO 73
88  83 READ(52)IN
89  9 IF(IN.GT.100)GO TO 10
90  10 WRITE(0,16)
91  16 FORMAT(*,20X,CLASSES OFFERED BY THIS DEPARTMENT AT SPECIFI
92  16 TIMES,*,37(*,**),)
93  160 CONTINUE
94  192 DO 230 II=1,N
95  230 IF(DAY.GT.9.K.DAY,KTIME,KTERM)
96  202 WRITE(0,12)KDAY,KTIME,KTERM
97  12 FORMAT(*,37(*,**),)" KEY PUNCH ERROR --カード メスは ?,312)
98  205 GO TO 230
99  205 IF(KDAY.GT.9.K.DAY,KTIME,KTERM)
100  25 WRITE(0,14)KDAY,KTERM
101  14 FORMAT(*,37(*,**),)
ALGORITHM FOR LOCATING RECORD FROM ABOVE CODES

101 IF (KDAY, EQ. 1) KCAY = 2
102 L = (KCAY - 1) * 10 + TIME * (TERM - 1) * 142

C READ CLASSES OFFERED AT THESE TIMES AND PRINT OUT

C

103 F FORMAT(2012)
104 READ(9*1, 13) (INK(K), K=1, 20)
105 IF (INK(K), EQ. 0) GO TO 220
106 DO 210 K = 1, 20
107 J = INK(K)
108 IF (J, EQ. 0) GO TO 230
109 WRITE(6, 15) (CLASS(J, K), K=1, 2)
110 15 FORMAT(*G1, 5X, 2A4)
111 220 WRITE(6, 1000)
112 1000 FORMAT(*G1, 3X, *NO CLASSES OFFERED BY THIS DEPARTMENT AT THIS TIME $*)
113 230 CONTINUE
114 WRITE(6, 7)
115 250 WRITE(6, 7)
116 STOP
117 END
APPENDIX C

ADVISOR

Source Program Listing
PROGRAM: ADVISOR

DATE: SEPT. 1971

AUTHOR: R. R. HEWETT

DESCRIPTION: ADVISOR IS A PROGRAM WRITTEN IN FORTRAN

Which has as its purpose the advising of

students as to what classes they should take

Advisor is a general program aimed at

the department level. It has been written

so that any department can use it for students

taking classes from them.

In order to use Advisor, the department

must make available to this program certain

data. This data is described and stored in

disk files by another program known as ACILIB

FILE DISCRIPITIONS:

FILE NO. 1 CLASS LIST FILE

The class list file contains the name, title,

class type (full, A or B half), and a synonym

code (which indicates a previous name for a class) for

each class offered by the department and any other

classes which are prerequisites.

FILE NO. 2 PREREQUISITE FILE

This file contains a logical condition code

(which indicates the arrangement of ANS and ORS)

and the sequence numbers of the prerequisites.

FILE NO. 3 REQUIRED CLASSES FILE

This file contains data that indicate the

number of colleges, the number of options for each

college, the options, and the required classes.

FILE NO. 4 INTEREST TOPIC COVERAGE FILE

This file contains the rating of each topic

for each class.

FILE NO. 5 TOPIC DESCRIPTION FILE

This file contains the description of each

topic, which is later printed out to the student

with his rating for each. It also contains the

number of topics for this department.

FILE NO. 14 TIME FILE INDEX

This file contains the locations of each class

in file 15.

FILE NO. 15 TIME FILE

This file contains the section numbers, times,

rooms and instructors for each class given by

the department.
COMMON NMAJ, NCOL
DIMENSION ITOT(50, 9), NOPT(10), NOUT(10, 3)
DIMENSION NOTOF(10, 3), KCREU(10, 10), NSTDU(50, 5)
DIMENSION ICVRK(50, 80), INTRST(80), NSRRT(50, 6), NTYPE(5)
DIMENSION NOFRN(10, NRC(10), NCT(24), NOUT(10), IPRINT(5)
DIMENSION NCOLGE(3, 4, NRE(12), TITLE(70, 15)
DIMENSION ISPACE(601, ITU(100), IVALU(5)
DEFINE FILE 15 (SOU, (O1, I15))
DATA NVALU/-NATE- NO. "SOME", "FAIR", "HIGH"/
DATA NCOLGE/ COMM, "ARTS", "EDUC", "ERE", "AND", "AYIL", " SCI"
DATA NREP/ NOT, */ DATA NTYPF/FULL," A", "S", "ORB","B"/
DATA INBLNK, ITX," X"
DATA VBLK, ITX, X
```
INTEGER A, A1, C1, C2

C
C*****************************************************************************
C REAC IN SEQUENTIAL FILES FROM DISC
C IATC APPROPRIATE ARRAYS
C*****************************************************************************
C..CLASS LIST FILE
NSYN=0
DO 10 1=1, 50
READ(1, 1, END=15) INCLASS(I, J), J=1, 4), (TITLE(I, K), K=1, 15)
C..CHECK IF SYNONYM, IF SO, INCREASE COUNT OF SYNONYMS
IF(INCLASS(I, 3), GT, 0) NSYN=NSYN+1
10 CONTINUE
C..Determine number of current classes = A
C..A1 IS NUMBER OF CLASSES IN THE FILE
  15 A1=(NSYN+1)
  A1=1-1
C..PREREQUISITE FILE
C
DO 20 I=1, 4
  20 READ2(2, NPRE(I, J), J=1, 5)
C
C..REQUIRES CLASSES FILE
C
C..READ IN NO OF COLLEGES
  READ3, 7INR0P
C..READ IN A C OF OPTIONS FOR EACH COLLEGE
  DO 30 I=1, NROP
    READ(3, 7J), NCPT(I)
    IF(INOPT(I), EQ, 0) G010 30
C..READ IN OPTION CODES
    KR=NOPT(I)
  DO 25 K=I-KR
  25 READ(3, 7INOUT(I, K), NOTOF(I, K))
C..READ IN OPTIONS
    KS=NOUTOF(I, K)
  25 READ3, 8(1KOPT(I, K), J=1, KS)
C..READ IN REQUIRED CLASSES FOR THIS COLLEGE
  30 READ3, 8((I, J), K=1, 10)
C..Determine no of required classes for each college
  DO 40 J=1, NRC
```
DO 35 K=1,10
IF(KREQ(J,K).EQ.0)GOTO40
35 CONTINUE
40 NREQ(J)=K-1
C
C...TIME INDEX FILE
L=A+1
DC 45 I=1,L
45 READ(14,711)TIDX(I)
C*******************************************************************************
C ENTRY POINT FOR SECOND AND SUBSEQUENT STUDENTS
C*******************************************************************************
C
C...READ IN STUDENT DATA TO BE ADVISED
C...NCOL IS COLLEGE "STUDENT" BELONGS TO
C...MAJ IS = 1 IF NOT MAJOR,
C... = 2 IF MAJOR.
C
50 READ(5,11111,NM1,NM2,NM3,NM4,NM5,NSTUD,NCOL,MAJ)
C...CHECK FOR END OF STUDENTS (BLANK CARD)
IF(INSTUD.NE.C)GOTO55
WRITE(6,2055)
STOP
C...INITIALIZF STATUS CODE, WEIGHTS, AND NBHALF ARRAYS TO ZERO
55 DC 65 I=1,A
NCOL=I+5I=6
DO 60 J=1,9
60 ITCNT(I,J)=0
DO 65 K=1,3
65 NBHALF(I,K)=0
C
C...PRINT FIRST PAGE
WRITE(6,20101)
WRITE(6,20203)
WRITE(6,20301)NM1,NM2,NM3,NM4,NM5,NSTUD
WRITE(6,20401)(NCOLGE(ICOL),J=1,4)
WRITE(6,20501)INREP(MAJ)
C...SKIP REQUIREMENTS PAGE FOR NONMAJOR
IF(MAJ.EQ.1)GOTO86
C...PRINT MAJOR REQUIREMENTS PAGE
WRITE(6,2055)
WRITE(6,20603)
WRITE(6,20701)(NCOLGEINCOL,J,J=1,4)
WRITE(6,20801)
C...PRINT REQUIRED CLASSES
K=KREQ(NCOL)
DO 70 I=1,K
LK=KREQ(NCOL+1)
LK2=NCOL+K
70 WRITE(6,20501)(NCOL(K),J,J=1,2),NCLASS(LK),NCLASS(LK+1),NCLASS(LK+2),NCLASS(LK+3)
C...PRINT REQUIRED CLASSES
IF(INOPT(NCOL).EQ.0)GOT77
K=NOPT(NCOL)
DO 77 J=1,K
77 WRITE(6,2100)NCOL(NCOL,J)
K=NOPT(NCOL+1)
DC75 J=1,K5
LK=KOPT(NCOL+1)
LK2=NCOL+K+1
75 WRITE(6,20901)(NCOL,(J,J=1,2),NCLASS(LK),NCLASS(LK+1),NCLASS(LK+2),NCLASS(LK+3)
C...
77 WRITE(6,1114)
80 WRITE(6,1112)
WRITE(6,1113) INI, NM2, NM3, NM5, NSTD
WRITE(6,9)
C...READ IN (CARDS) CLASSES TAKEN BY STUDENT
C...FIRST CARD INDICATES NUMBER OF CLASSES
READ(5,7)NCL
IF(INCL.EQ.0)GO TO 93
DO92 I=1,NCL
READ(5,5000)CI,C2
WRITE(6,5010)CI,C2
C...CHECK IF CLASS IS IN CLASS LIST FILE
DO90 J=1,A1
IF(INCLASS(J,1).NE.C1.OR. NCLASS(J,2).NE.C2)GOTOC90
C...CHECK IF CLASS IS A SYNONYM
IF(INCLASS(J,3).GT.0)GOTOC85
C...SET STATUS CODE TO TAKEN
NCLASS(J,5)=1
GOT092
C...SET STATUS CODE OF PARENT CLASS TO TAKEN
85 KK=NCLASS(J,3)
NCLASS(KK,5)=1
GOT092
90 CONTINUE
92 CONTINUE
C...CHECK IF STUDENT HAS SPECIAL PERMISSION TO TAKE ANY CLASSES
C
93 READ(5,7)NCLP
IF(INCLP.EQ.0)GO TO 97
WRITE(6,5011)
5011 FORMAT(24X,'SPECIAL PERMISSION HAS BEEN OBTAINED TO TAKE THE F
FOLLOWING CLASSES:"
DO 96 I=1,NCLP
READ(5,5000)CI,C2
WRITE(6,5010)CI,C2
DO 94 J=1,A1
IF(INCLASS(J,1).NE.C1.OR. NCLASS(J,2).NE.C2)GO TO 94
IF(INCLASS(J,5).EQ.1)GOTOC96
NCLASS(J,5)=NCLASS(J,4)+2
94 CONTINUE
96 CONTINUE
97 REWIND 8
C...READ FROM DISC NO OF INTEREST TOPICS
C
C...READ FROM DISC STUDENT'S INTEREST RATINGS
REWIND 8
READ(8,7)INTOP
READ(5,5020)INTRST(J,J=1,INTOP)
C...PRINT OUT INTEREST TOPICS AND THIS STUDENT'S RATINGS
WRITE(6,5050).
DO 95 I=1,INTOP
READ(8,5030)ISPACE(J,J=1,12)
K=INTRST(J)+1
95 WRITE(6,5040)ISPACE(J,J=1,12),K,IVALU(K)
C...READ IN INTEREST COVERAGE FILE
REWIND
DO 100 J=1,A
100 READ(4,1100)ICVRG(J,K=1,INTOP)
C意识到连续的需要
UPDATE STATUS CODE=1 FOR TAKEN CLASSES

=2 FOR POTENTIAL FULL CLASSES
=3 FOR POTENTIAL "A" HALF CLASSES

=4 FOR POTENTIAL "B" HALF CLASSES
=5 FOR POTENTIAL "A OR B" HALF CLASSES
=6 POTENTIAL "B" CLASS WITH POTENTIAL "A" AS PREREQ

POTENTIAL FULL CLASSES

FINE CLASS TYPE=0 AND STATUS CODE=0
DO 110 L=1,A
IF(INCLASS(L,4).NE.0 OR NCLASS(L,5).NE.0)goto110

CHECK IF PREREQUISITES HAVE BEEN SATISFIED
CALL PRECK(L,1,NORY)
IF(NORY.EQ.0)INCLASS(L,5)=2
IF(NORY.EQ.2)INCLASS(L,5)=1
110 CONTINUE

POTENTIAL "A" HALF CLASSES

FINE CLASS TYPE=1 AND STATUS CODE=0
DO 120 L=1,A
IF(INCLASS(L,4).NE.1 OR NCLASS(L,5).NE.0)goto120
CALL PRECK(L,1,NORY)
IF(NORY.EQ.0)INCLASS(L,5)=2
IF(NORY.EQ.2)INCLASS(L,5)=1
120 CONTINUE

POTENTIAL "B" HALF CLASSES

FINE CLASS TYPE=2 AND STATUS CODE=0
DO 125 L=1,A
IF(INCLASS(L,4).NE.2 OR NCLASS(L,5).NE.0)goto125
CALL PRECK(L,2,NORY)
IF(NORY.EQ.0)INCLASS(L,5)=2
IF(NORY.EQ.2)INCLASS(L,5)=1
125 CONTINUE

POTENTIAL "A OR B" HALF CLASSES

FINE CLASS TYPE=3 AND STATUS CODE=0
DO 145 L=1,A
IF(INCLASS(L,4).NE.3 OR NCLASS(L,5).NE.0)goto145
CALL PRECK(L,2,NORY)
IF(NORY.EQ.1135)I4G,130
130 NCLASS(L,5)=1
135 NCLASS(L,5)=5
GO TO 145

CHECK TO SEE IF POTENTIAL "B" HALF CLASS
140 CALL PRECK(L,2,NORY)
IF(NORY.EQ.0)INCLASS(L,5)=4
IF(NORY.EQ.2)INCLASS(L,5)=1
IF(NORY.EQ.3)INCLASS(L,5)=6
145 CONTINUE
C PRINT INSTRUCTIONS FOR USING ADVISOR

C******************************************************************************
WRITE(6,4000)
WRITE(6,4010)
WRITE(6,4020)
WRITE(6,4030)
WRITE(6,4040)
C******************************************************************************
C WEIGHTING FOR REQUIRED CLASSES
C******************************************************************************
C...CHECK TO SEE IF MAJOR --- IF NOT BRANCH TO INTEREST-COVERAGE MATCH
IF(NMAJ.EQ.1) GO TO 220
C...NR1 IS THE NUMBER OF REQUIRED CLASSES FOR THIS COLLEGE
NR1=NR4(NCOL)
DC 215 J=1,NR1
C...GET STATUS CODE FOR REQUIRED CLASS
L=KREQ(NCOL,J)
K=CLASS(L,5)
C...ANALYZE STATUS CODE --- IF TAKEN THEN GET NEXT Req'D CLASS
C...	IF(K-I)150,215,210
C...	IF SENIOR CHECK PREREQUISITES
C...	IF POTENTIAL ADD WEIGHT FOR REQUIRED CLASS
C...
C...CHECK IF PREREQUISITES ARE POTENTIAL
C...
190 K1=1
L1=1
CALL RECPRE(L1,N1,K1)
DC 205 J1=1,N1
IF(KODE(K1,J1)-1)1205,155,160
C...ADD WEIGHT TO PREREQUISITE FOR BEING A PREQ FOR A REQUIRED CLASS
155 I TotWT(M(K1,J1),3)=ITCTWT(M(K1,J1),3)+1500
ITOTWT(M(K1,J1),1)=ITCTWT(M(K1,J1),1)+1500
GO TO 205
C...SECOND LEVEL OF PREREQUISITES (I.E. PREREQUISITES OF PREREQUISITES)
160 L2=M(K1,J1)
K2=2
CALL RECPRE(L2,N2,K2)
DO 200 J2=1,N2
IF(KODE(K2,J2)-1)1200,165,170
165 I TotWT(M(K2,J2),3)=ITCTWT(M(K2,J2),3)+1500
ITOTWT(M(K2,J2),1)=ITCTWT(M(K2,J2),1)+1500
GOTO 200
C...THIRD LEVEL OF PREREQUISITES
170 L3=M(K2,J2)
K3=3
CALL RECPRE(L3,N3,K3)
DO195 J3=1,N3
IF(KODE(K3,J3)-1)195,175,180
175 I TotWT(M(K3,J3),3)=ITCTWT(M(K3,J3),3)+1500
ITOTWT(M(K3,J3),1)=ITCTWT(M(K3,J3),1)+1500
GOTO195
C...FOURTH AND FINAL LEVEL OF PREREQUISITES
180 L4=M(K3,J3)
K4=4
CALL RECPRE(L4,N4,K4)
DC 190 J4=1,N4

IF (IKODE(K4, J4) EQ 1) ITOT = ITOT + 150, 185, 190
185 ITOTWT(K4, J4, 3) = ITOTWT(K4, J4, 3) + 1500
ITOTWT(K4, J4, 3) = ITOTWT(K4, J4, 3) + 1500
190 CONTINUE
195 CONTINUE
200 CONTINUE
205 CONTINUE
GOTO 215
C ... ADD WEIGHT FOR POTENTIAL REQUIRED CLASS
210 ITOTWT(L, 2) = ITOTWT(L, 2) + 1500
ITOTWT(L, 1) = ITOTWT(L, 1) + 1500
215 CONTINUE
220 ITOT = 0
225 DO 238 I = 1, A
230 IF (ICVRGL, I, EQ 0) GO TO 239
DC 235 K = 1, NTOP
TNATE = 0
L = INTRST(K)
J = ICVRGL(I, K)
GO TO [235, 221, 226, 230, J]
221 TNATE = 10
IF (L.EQ.0) TNATE = 25
IF (L.EQ.1) TNATE = 10
GO TO 234
226 TNATE = 20
IF (L.EQ.0) TNATE = 50
IF (L.EQ.1) TNATE = 20
IF (L.EQ.2) TNATE = 10
GO TO 234
230 IF (L.EQ.3) TNATE = 30
IF (L.EQ.0) TNATE = 75
IF (L.EQ.1) TNATE = 10
IF (L.EQ.2) TNATE = 20
224 ITOTWT(I, 1) = ITOTWT(I, 1) + TNATE
235 CONTINUE
IF (L.EQ.0) ITOTWT(I, 1) = ITOTWT(I, 1) + TNATE
ITOT = ITOT + ITOTWT(I, 1)
238 CONTINUE
I = A
239 AVERAGE = ITOT/(10*1.0)
C
C CHECK IF MAJOR
C
IF (NOT, BRANCH TO WEIGHTING FOR INTEREST PREREQS FOR NON MAJORS
IF (INMAJ, EQ 1) GO TO 255
K1 = NOPT(NCOL)
IF (NOPT(NCOL), EQ 0, JK1 = 1
DO 350 K4 = 1, K1
DO 350 J = 1, A
KCP = 0
IF (NOPT(NCOL), EQ .0, JK1 = 1
DC 240 K3 = 1, K2
IF (L.EQ.0, KOP = 1, KOP = 1, KOP = 1
KOP = 1
GO TO 245
245 IF (CLASS(I, 5) EQ .1, GO TO 250
IF (CLASS(I, 1) EQ .1, GO TO 250
ITOTWT(I) = ITOTWT(I) + ITOTWT(I) + ITOTWT(I)
Go to 300

250 IF (TOT(T(I, 6)) .LT. IAVG) GC TO 265
C CHECK IF PREREQ HAS POTENTIAL STATUS
C
K = 1
CALL RECPR(1, N, K)
DO 260 J = 1, N
IF (CODE(K(J)) = 1) 260, 255, 260
255 TOT(K(J), 1) = TOT(K(J), 1) + TOT(T(I, 6))
1OT(K(J), 1) = TOT(K(J), 1) + TOT(T(I, 6))
C CONTINUE
260 CONTINUE
265 IF (KDP.EQ.0) 260 TOT(T(I, 6)) = 0
300 CONTINUE
IF (INOPT(INCOL = 0)) 250, 320, 300
C WEIGHT FOR REQUIRED OPTIONS
C
KC = NOUT-CN, K+1
KN = NOUT-CN, K+1
DO 302 I = 1, KN
302 NOPTN(I) = NOPTN(I)
DO 304 I = 1, KN
N = NOPTN(I)
IF (NNCLASS(N, 5) .EQ. 0) GO TO 308
306 CONTINUE
GO TO 320
308 KD = KO - 1
IF (KD .EQ. 0) GO TO 350
IF (I .EQ. KN) GO TO 312
KN = KN - 1
DO 310 J = 1, KN
310 NOPTN(J) = NOPTN(J + 1)
GO TO 304
312 KN = KN - 1
320 DC 323 I = 1, KA
323 TOT(I) = TOT(T(NOPTN(I), 6))
JSHT = 0
DO 325 I = 1, KA
325 K = I + 1
DO 325 J = K, KN
IF (NTOT(J) .GE. NTOT(I)) GO TO 325
ITEMP = NTOT(I)
NTOT(I) = NTOT(J)
NTOT(J) = ITEMP
325 CONTINUE
DO 326 I = 1, KA
326 N = NOPTN(I)
IF (NNCLASS(N, 5) .EQ. 0) GO TO 327
1OT(T(N+1)) = 1500
1OT(K(N+1)) = 1500
KO = KO - 1
327 K = 1
LST*T = 0
CALLREQPRE(I,N,K)
DC 340 L=1,J
IF(KODE(K,K,J)=1)340,335,340
335 IF(I<ITOTW(I(M(K,K,J),I),G-0)GO TO 340
ITOTW(I(M(K,K,J),I),G-1500
ITOTW(I(M(K,K,J),I)+1500
LSW1=1
KLAST=1
340 CONTINUE
IF(LSW1.EQ.0)GO TO 330
JSH1=JSH1-1
K0=K0-1
IF(KO332,332,330
330 CONTINUE
332 IF(JSH1.EQ.0)GO TO 350
J=KLAST
DO 345 L=1,JSH1
J=J+1
IF(J.GT.KN)GO TO 350
N=NOPTN(J)
IF(INCLASS(N,51.EQ.0)GO TO 345
ITOTW(N,8)=1500
345 CONTINUE
350 CONTINUE
GO TO 380
C WEIGHTING FOR INTEREST PREREQS FOR NON MAJORS
C 355 DO 378 I=1,A
IF(INCLASS(I,51.EQ.1)GO TO 378
IF(INCLASS(I,51.NE.0.IGO TO 375
IF(I<ITOTW(I,6).LT.IAVRGO TO 370
C CHECK TO SEE IF ABOVE AVERAGE INTEREST SENIOR CLASS HAS PREREQS THAT ARE
C POTENTIAL
K=1
CALL REQPRE(I,N,K)
DO365 J=1,N
IF(KODE(K,J)=1)365,360,365
360 I<ITOTW(I(M(K,J),I)+ITOTW(I(M(K,J),I)+ITOTW(I,J)
ITOTW(I(M(K,J),I)=ITOTW(I(M(K,J),I)+ITOTW(I,J)
365 CONTINUE
370 I<ITOTW(I,6)=0
GO TO 378
375 I<ITOTW(I,1)+ITOTW(I,1)+ITOTW(I,6)
378 CONTINUE
C SUCCESSION OF CLASSES
C CSORT CLASSES IN ORDER OF TOTL WEIGHTS
C 380 KTIMES=0
WRITE(6,4050)
4050 FORMAT('I',38X, 'SCHEDULE 1. IF YOU ARE PLANNING TO TAKE ADDITION
9 CLASSES',//,47X, 'AFTER THE COMING YEAR*)
390 DO 400 L=1,A
NSORTIL<L,1=
400 NSORTL=NSORTILJ+NCCLASS(L,J)
410 JJ=A-1
   DO 500 I=1, JJ
   J=I+1
   DO 500 K=J, A
   IF (IOTOT(K,1).LE.IOTOT(I,1)) GO TO 500
   DO 425 L=1, 6
   ITEM = NSORT(I,L)
   NSORT(I,L)=NSCRT(K,L)
425
   ITEM=IOTOT(I,L)
   IOTOT(I,L)=ITEM
   DC 450 L=1, 9
   IF (ITEM.EQ.1) NSORT(I,L)=ITEM
450
   CONTINUE
   WRITE(6,180) NK1, NM1, NM2, NM3, NM4, NM5, NSTUD
C
   SUGGEST CLASSES
C
   DO 508 I=1, 7
508  IPRINT(I)=IBLAK
   IB=0
   IF (NM1.EQ.2 .AND. KTIMES.EQ.0) GO TO 525
   IF (NM1.EQ.1 .AND. KTIMES.EQ.0) GO TO 535
   IF (NM1.EQ.2 .AND. KTIMES.EQ.1) GO TO 525
   IF (NM1.EQ.1 .AND. KTIMES.EQ.1) GO TO 545
505  WRITE(6,1500)
1500 FORMAT (-14,37A3, 'REASONS FOR BEING SUGGESTED')
   WRITE(6,1510)
   WRITE(6,1520)
   WRITE(6,1530)
   WRITE(6,1540)
   DO 521 J=1, A
   IF (IOTOT(I,J).NE.0) GO TO 521
520  DO 509 J=1, 9
509  IPRINT(J)=IBLAK
   IF (IOTOT(I,J).NE.0) IPRINT(J)=IX
   IF (IOTOT(I,J).NE.0) IPRINT(J)=II
   IF (IOTOT(I,J).NE.0) IPRINT(J)=IX
   IF (IOTOT(I,J).NE.0) IPRINT(J)=IX
   IPRINT(J)=IX
   N=IOTOT(I,J)
   IPRINT(N)=CLASS(N,1)
   IPRINT(N)=CLASS(N,2)
510  IF (NSCRT(I,J).NE.0) IPRINT(J)=IX
   N=NSORT(I,J,5)-1
   IF (NSCRT(I,J,5).NE.0) IPRINT(J)=IX
   IB=IB+1
   WRITE(6,2110) IB
   GO TO 520
515  WRITE(6,2120)
520  WRITE(6,2000) NSORT(I,J), IJ=1, 2), NTYPE(N), IOTOT(I,J), IPRINT(K), K=
      1, 6), IPRINT(J), IPRINT(J)}
DC 531 I=1,A
IF(110T(T1,1) .EQ. 0) GO TO 531
DO 526 J=1,9
526 I PRINT(J)=IBLNK
IF(110T(T1,2).NE.0) IPRINT(1)=1X
IF(110T(T1,4).NE.0) IPRINT(2)=1X
N=NSORT(I,5)-1
IF(NSORT(I,5).NE.6) GO TO 528
IB=18+1
WRITE(6,3040)IB
GO TO 530
528 WRITE(6,2120)
530 WRITE(6,3010)(NSORT(I,J),J=1,2),NTYPE(N),I10TWT(1,1),IPRINT(1),IPRINT(2),I10TWT(1,6)
531 CONTINUE
GO TO 625
535 WRITE(6,1705)
WRITE(6,1710)
WRITE(6,1720)
WRITE(6,1730)
WRITE(6,1740)
DO 543 I=1,A
IF(110T(T1,1).EQ.0) GO TO 543
DO 536 J=1,9
536 I PRINT(J)=IBLNK
IF(110T(T1,2).NE.0) IPRINT(1)=1X
N=NSORT(I,5)-1
IF(NSORT(I,5).NE.6) GO TO 540
IB=18+1
WRITE(6,3030)IB
GO TO 542
540 WRITE(6,2120)
542 WRITE(6,3000)(NSORT(I,J),J=1,2),NTYPE(N),I10TWT(1,1),I10TWT(1,6),I10TWT(1,8),1
IPRINT(J)
543 CONTINUE
GO TO 625
545 WRITE(6,1760)
WRITE(6,1770)
WRITE(6,1780)
WRITE(6,1790)
DC 566 I=1,A
IF(110T(T1,1).EQ.0) GO TO 566
DO 546 J=1,9
546 I PRINT(J)=IBLNK
N=NSORT(I,5)-1
IF(NSORT(I,5).NE.6) GO TO 550
IB=18+1
WRITE(6,3050)IB
GO TO 565
550 WRITE(6,2120)
555 WRITE(6,3020)(NSORT(I,J),J=1,2),NTYPE(N),I10TWT(1,1),I10TWT(1,6)
566 CONTINUE
625 IF(I18.EQ.0) GO TO 700
N=0
DC 650 I=1,A
IF(N.GT.16) GO TO 700
IF(NSORT(I,5).NE.6) GO TO 650
N=N+1
550 WRITE(6,2130)
DO 650 J=1,3
     IF(NHALF(L,J).EQ.0) GO TO 650
     K=NHALF(L,J)
     WRITE(6,2140)(INCLASS(K,LI),LI=1,2)
   650 CONTINUE
   2130 FORMAT('**',20X,'NOTE',13,'THE FOLLOWING A HALF CLASSES MUST BE TAKEN BEFORE THIS B HALF CLASS:')
   2140 FORMAT('**',1X,2A4)
   700 TIMES=K TIMES+1
     WRITE(6,6000)
   6000 FORMAT('**',1X,45X,'TIMES AND INSTRUCTORS OF THE TOP TEN RANKING CLASSES AND TOP TEN RANKING INSTRUCTORS FROM ABOVE ARE GIVEN ON THE FOLLOWING PAGE:')
     WRITE(6,6010)
   6010 FORMAT('**',1X,45X,'TIMES AND INSTRUCTORS OF TOP TEN CLASSES:')
     DO 751 I=1,10
       J=NSORT(I,16)
       K=NSORT(I,51)-1
       WRITE(6,6020)INCLASS(J,1),INCLASS(J,2),NTYPE(K,J),(TITLE(J,L),L=1,15)
       IF(NTOTW(I,J).EQ.0) GO TO 751
       WRITE(6,6030)INCLASS(J,3),INCLASS(J,4),INTIME(J,K),INCLASS(J,5)
       WRITE(6,6040)INCLASS(J,6),INCLASS(J,7),INCLASS(J,8),INCLASS(J,9),INCLASS(J,10)
       WRITE(6,6050)INCLASS(J,11),INCLASS(J,12),INCLASS(J,13),INCLASS(J,14),INCLASS(J,15)
       WRITE(6,6060)INCLASS(J,16)
     CONTINUE
   751 CONTINUE
   752 FORMAT('**',1X,2A4,10X,1X,45X,'THE FOLLOWING A TOP TEN RANKING COURSES ARE GIVEN ON THE FOLLOWING PAGE:')
     WRITE(6,7530)
   7530 FORMAT('**',1X,45X,'THE FOLLOWING A TOP TEN RANKING COURSES ARE GIVEN ON THE FOLLOWING PAGE:')
     DO 800 J=1,10
       WRITE(6,8010)INCLASS(J,1),INCLASS(J,2),INCLASS(J,3),INCLASS(J,4),INCLASS(J,5)
       WRITE(6,8020)INCLASS(J,6),INCLASS(J,7),INCLASS(J,8),INCLASS(J,9),INCLASS(J,10)
       WRITE(6,8030)INCLASS(J,11),INCLASS(J,12),INCLASS(J,13),INCLASS(J,14),INCLASS(J,15)
       WRITE(6,8040)INCLASS(J,16)
     CONTINUE
   800 CONTINUE
FOR POSSIBLY
FORMAT [$]

CAN

THE

THAT

ADVISOR’S

FORMAT [$]

FORMAT [$]

UPEU"

INTERNET

INTERNET

RE", "RATING", "CLASS", "INTEREST"

5X, 121=r", 5X, 121=r", 5X, 121=r", 5X, 121=r")

FORMAT [$]

FORMAT [$]

FORMAT [$]

FORMAT [$]

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR

FOR
$ASS, * // T20, * WHEN SELECTING CLASSES FROM THIS LIST TO FIT YOUR
$TIME TABLE, IT IS IMPORTANT THAT YOU ATTEMPT TO TAKE*, //
$ELE, If YOU WISH TO TAKE FOUR CLASSES FROM THIS DEPARTMENT, * //,
$T20, * YOU SHOULD SELECT THE TOP FOUR IN THE SCHEDULE. IF FOREVER,
$ ONE OF THESE CLASSES CANNOT FIT INTO* // T20, * YOUR TIME TABLE,
$ THEN YOU SHOULD SUBSTITUTE THE FIFTH HIGHEST RATED CLASS IN THE L
$IST, AND SO ON.*

4030 FORMAT(*-T10, 'SCHEDULE 1', // T25, 'THIS SET OF CLASSES IS MEANT'
$ FOR THE STUDENT WHO WILL BE TAKING ADDITIONAL CLASSES FROM THIS*
$*, // T20, DEPARTMENT AFTER THE COMING YEAR,* // T25, * IT IS IMPO-
$RANT THEN, THAT THE PREREQUISITES FOR SENIOR CLASSES ARE SATISFIE
$D NEXT YEAR.*, // T20, * SO THAT THE SENIOR CLASSES MAY BE TAKEN IN
$THE FOLLOWING YEAR.* // T25, * THIS SCHEDULE, THEREFORE, GIVES A H
$IGHER RATING TO THOSE CLASSES THAT ARE PREREQUISITES FOR* // T3
$0, *. A REQUIRED CLASS, IN THE CASE OF A MAJOR,* // T30, '2* A
$REQUIRED OPTION, IN THE CASE OF A MAJOR.* // T30, '3* A SENIOR CL
$ASS IN WHICH THE STUDENT HAS AN ABOVE AVERAGE INTEREST.*

4040 FORMAT(*-T10, 'SCHEDULE 2', // T25, 'THIS SET OF CLASSES IS MEANT'
$ FOR THE STUDENT WHO WILL NOT BE TAKING ADDITIONAL CLASSES*, //, T
$T20, * FROM THIS DEPARTMENT AFTER THE COMING YEAR,* // T25, * THEREFO
$RE, ONLY THOSE CLASSES THAT CAN BE TAKEN IN THE COMING YEAR WILL
$BE CONSIDERED.*

END

SUBROUTINE REGPRE(L, N, K)
COMMON KODE(4, 4), M4(4, 4), NCLASS(10, 5), NPREQ(50), NBHALF(5, 3)
COMMON 'NMAJ, NCOL
DIMENSION JJ(40), NN(4), KK(4), MM(4)
DO 20 I=1, 1
IF(INPREQ(I, I).EQ.0) GO TO 25
JJ(I)=NPREQ(I, I)
NN(I)=NCLASS(JJ(I), 5)
20 CONTINUE
25 LGP=APREQ(I, 1)
GO TO (30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 30, 30, 30) LGP
30 KODE(K, I)=1
IF(NN(I).LT.2) KODE(K, I)=2
N=1
M(K, N)=JJ(I)
RETURN
40 N=2
41 DO 45 I=1, N
KODE(K, I)=1
IF(NN(I).LT.2) KODE(K, I)=2
45 M(K, I)=JJ(I)
RETURN
50 N=2
51 J=1
52 DO 55 I=J, N
KODE(K, I)=0
53 KODE(K, I)=2
GO TO 55
54 KODE(K, I)=1
55 M(K, I)=JJ(I)
RETURN
60 N=3
60 GO TO 41
70 N=3
70 GO TO 51
80 N=4
GO TO 51
90 N=4
GO TO 41
100 N=4
IF(N1(1).EQ.1.OR.NN(2).EQ.1) GO TO 105
DO 102 I=1,2
KODE(K,I)=1
IF(N1(1).LT.2)KODE(K,I)=2
102 M(K,I)=J(J(1))
GO TO 107
105 DO 106 I=1,2
106 KODE(K,I)=0
107 J=3
GO TO 52
110 N=3
IF(N1(1).EQ.1.OR.NN(2).EQ.1) GO TO 115
DO 112 I=1,2
KODE(K,I)=1
IF(N1(1).LT.2)KODE(K,I)=2
112 M(K,I)=J(J(2))
GO TO 117
115 DO 116 I=1,2
116 KODE(K,I)=0
117 J=3
GO TO 52
120 N=4
IF(N1(1).EQ.1.OR.NN(2).EQ.1.OR.NN(3).EQ.1) GO TO 125
DO 122 I=1,3
KODE(K,I)=1
IF(N1(1).LT.2)KODE(K,I)=2
122 M(K,I)=J(J(1))
GO TO 127
125 DC 126 I=1,3
126 KODE(K,I)=0
127 J=6
GO TO 52
SUBROUTINE PRECK (L,K,NDRY)
COMMON KODE (4,4), L(4,4), NCLASS (70,5), NPREQ(50,5), NBHALF (50,3)
COMMON NMAJ,NCOL
INTEGER A
DIMENSION A(4)
IF(NPREQ(L,1).EQ.0) GO TO 13
IF(NPREQ(L,1).EQ.13) GO TO 90
IF(NPREQ(L,1).EQ.14) GO TO 90
DO 20 I=1,4
IF(NPREQ(L,1+1).EQ.0) GO TO 25
J=1
20 A(I)=NCLASS(NPREQ(L,1+1),5)
25 N=0
DO 30 I=1,J
IF(A(I).EQ.1)N=N+1
30 CONTINUE
IF(N=0) GO TO 13
GO TO (1,1,2,1,2,1,2,1,8,9,11,12),LOG
1 IF(N,GT.0) GO TO 13
2 IF(K,EQ.1)GO TO 14
3 LL=0
4 DC 35 I=1,J
IF(A(1).NE.3) GO TO 35
LL=LL+1
NBHALFIL,LL1=NPREQIL,141
GO TO 14
35 CONTINUE
GO TO 14
2 IFIN.EQ.J GO TO 13
IFK.EQ.I1GO TO 14
LL=0
DO40I=1,J
IF(A(I).NE.3) GO TO 40
LL=LL+1
NBHALFIL,LL1=NPREQIL,141
N=N+1
IFIN.NE.J GO TO 40
GO TO 16
40 CONTINUE
GO TO 14
8 IF(A(1).NE.1.OR.A(4).NE.1 GO TO 41
IF(A(1).EQ.1.CR.A(2).EC.1 GO TO 13
41 K1=2
K2=4
42 IFK.EQ.1 GO TO 14
DO 45 I=1,K1
IF(A(I).EQ.1) GO TO 60
45 CONTINUE
LL=0
DO 50 JJ=1,K1
IF(A(JJ).NE.3) GO TO 50
LL=LL+1
NBHALFIL,LL1=NPREQIL,141
50 CONTINUE
IFILL-1)14,55,55
55 N=N+1
IFIN.EQ.J GO TO 16
60 LL=0
K3=K1+1
DO 70I=1,K3,K2
IF(A(I).NE.3) GO TO 70
LL=LL+1
NBHALFIL,LL1=NPREQIL,141
N=N+1
IFIN.EQ.1 GO TO 16
70 CONTINUE
GO TO 14
9 IF(A(3).NE.1) GO TO 71
IF(A(1).EQ.1.OR.A(2).EQ.1 GO TO 13
71 K1=2
K2=3
GO TO 42
10 IF(A(4).NE.1) GO TO 72
IF(A(1).EQ.1.OR.A(2).EQ.1.OR.A(3).EQ.1) GO TO 13
72 K1=3
K2=4
GO TO 42
80 IFINMAG.EQ.2) GO TO 15
J=NPREQIL,2)
IFINCCLASSJ,5F.EQ.0 GO TO 14
IFINCCLASSJ,5).EQ.3 GO TO 16

GO TO 13
90 J=NPREG(L+2)
   IF(NMAJ.EQ.2.AND.NCOL.NE.J)GO TO 15
   GO TO 95 I=1,J
   J=NPREG(L+2)
   IF(NCLASS(J,5).EQ.0)GO TO 95
   IF(NCLASS(J,5).EQ.3)GO TO 16
   GO TO 13
95 CONTINUE
   GO TO 14
   11 IF(A(2).EQ.1.OR.A(3).EQ.1)GO TO 15
   IF(A(1).EQ.1)GO TO 13
   IF(A.2).EQ.1)GO TO 14
   IF(A(1).EQ.3)GO TO 16
   GO TO 14
12 IF(A(1).EQ.1.OR.A(2).EQ.1)GO TO 15
13 NCRY=0
   RETURN
14 NORY=1
   RETURN
15 NORY=2
   RETURN
16 NCRY=3
   RETURN
END

/*