ADVISER - A PROGRAM WHICH ADVISES STUDENTS ON COURSES

Eric M. Timmreck

Computer Sciences Technical Report #16
February 1968

ADVISER - A Program which Advises Students on Courses

Abstract

A computer program which advises students on Computer Sciences courses has been used experimentally at the University of Wisconsin. The program takes into account the background of the student, his interests, his degree requirements, possible schedule conflicts, similarities between courses, and other factors. The student can disagree with the evaluations made by the program and can to some extent direct it in its search for good sets of courses. Of eleven students who participated in the experiment, ten were in general satisfied with the results of the program and eight felt that it performed as well as the average human adviser.

I. Objective

A program to advise students at the University of Wisconsin concerning Computer Sciences courses has been written and experimentally used. The program is written in B5500 Algol and runs on the B5500 with remote teletypes.

The goal here was to include in a program as many as possible of the features of a human student adviser. It seemed that in certain areas the program would be able to surpass the average human, such as in thoroughness in investigating the student's background and in knowledge of the content of all the courses offered by the Computer Sciences Department. Certain other areas would not be amenable to computer handling, such as the use of grapevine information concerning courses and the recognizing of human personality traits which might influence the speed at which a student is to progress. In spite of these limitations, it was believed that a program could do a good job of advising in a very large number of cases, though to be safe the student would then want a human adviser to check over and supplement the program's output.

The program has, in fact, demonstrated that it can do good advising. Results will be discussed in detail in Section III.

Though many parts of the ADVISER program are specific to the particular department for which it was written, its general concepts and, in fact, large pieces of code could be used with only minor modifications for other academic departments. This will be discussed in detail in Section IV.

II. The Program

A. General description

The program is contained on about 3000 cards containing Algol statements. It is composed of about 20 major procedures and a number of minor ones. The two main sections of the program involve (1) interviewing the student to determine his background and interests, and (2) weighting and sorting the courses and obtaining the student's reactions to suggested course lists.

B. Data

Data for the program, contained on about 2200 cards, is divided into five files (accessed from disk): ADVISER/TOPICS, ADVISER/PREREQS, ADVISER/CDESCS, ADVISER/TIMES, and ADVISER/COURSES.

ADVISER/TOPICS (See Fig. 1) contains a list of 304 topics which are in some measure covered by Computer Sciences courses. The list is reasonably comprehensive but not totally exhaustive. It also contains in some cases more than one item meaning basically the same thing, but this probably improves the quality of the advice more than it hurts it. The list is somewhat hierarchical, having as its main headings

I. General information concerning the computing milieu

II. Numerical analysis

III. Systems programming

IV. Theory of computing

V. Artificial intelligence and related subjects.

Each of these is divided into subheadings, and some of those are in turn divided. The program allows this arrangement to go 6 levels deep, but only 4 levels are utilized by the current set of topics. The list was compiled from the list of topics used in Computing Reviews, faculty suggestions, and general knowledge about the departmental courses.

ADVISER/PREREQS (See Fig. 2) contains a list of all prerequisites for any departmental course which are not themselves departmental courses. This file is used for printout purposes when attempting to determine whether or not the student has satisfied the prerequisites for a particular course.

ADVISER/CDESCS contains for each course a verbal description of the course. It may contain more than one description for the same course, the later one being more detailed and complete. The program uses this latter option to describe a course more thoroughly if the student should ask for a verbal description of the course more than once.

ADVISER/TIMES (See Fig. 3) contains a list of all time periods during which courses are offered, having for each time period a number, an alphabetic description, and a list of all other time periods (by number) with which the one represented overlaps. This file is used for printout purposes and to insure that a suggested set of courses can be taken simultaneously without a time conflict.

ADVISER/COURSES (See Fig. 4) contains a large amount of specific information about the courses as well as the number of courses currently in the file. It contains a six-character designation (e.g. CS 132) and a 65-character description for each course. Prerequisites are also represented here. Each prerequisite is represented by a number. Numbers less than 60 refer to other departmental courses (which are numbered serially from the front end of ADVISER/COURSES); numbers above 60 refer to the prerequisites contained on the ADVISER/PREREQS file. A typical prerequisite representation might look like

14 15 16 AND 7 AND 92.

When one number is followed by another, an "OR" is assumed to exist between them. The "AND"s are considered the main connectives. The entire construct is ORed with "CONSENT OF INSTRUCTOR." In the example above, the prerequisite is considered satisfied if the student has taken (or equivalently taken) course 14 or 15 or 16 and course 7, and has satisifed prerequisite 92 (which might represent, for example, Math 223 - Calculus and Analytic Geometry). It is also considered satisfied if the student has obtained the consent of the instructor to take the course. A maximum of 12 elements (numbers and "AND"s) may be used to describe the prerequisites of a course.

The file ADVISER/COURSES also contains a 1-character designator to indicate the general type of course as follows:

I - Introductory course

A - Area A course (Numerical analysis)

B - Area B course (Systems programming and the theory of computing)

C - Area C course (Artificial intelligence and related subjects)

The number of credits which the course consists of is also contained, as are the date of last change on the data for the course, a list of times (by number) at which the course is to be offered (using -1 if course is not being offered and -2 if course is being offered but time has not yet been announced), the most current instructor for the course, a list of other departments in which the course can be taken for credit, and a flag to indicate whether of not the course involves quiz sections.

Finally, ADVISER/COURSES contains a digital description of the content of each course. Each course description contains two digits for each topic, one to indicate the amount of coverage of the topic by the course, the other to indicate the level of coverage of the topic by the course. The digits and their heuristically chosen meanings are as follows:

Amount	Level
0 - no coverage	0 - none
1 - a little	1 - elementary
2 - a fair amount	2 - intermediate
3 - a great deal	3 - advanced

These two aspects (amount and level) and their meanings are by no means well-defined, but they have served well to date. The digital course descriptions are obtained by passing out a form for each course to the instructor who is to teach it. The form contains all 304 topics, with two blanks by each (one for amount, one for level). The instructor is asked to fill in both blanks for each topic covered and to simply skip over any topics not covered. This is facilitated by arranging the topics on the form in hierarchical fashion, so that for any topic which is not covered, the instructor can skip all of its subtopics easily. It seems to take about 10 or 15 minutes to fill out this form, provided the instructor has already decided what he basically intends to cover.

Two other files declared in the program have not been used as yet. One contains records concerning all students advised; the other contains verbal descriptions of the topics. In a fully operational system, it might be desirable to use these, though it would by no means by necessary.

Collection of the initial set of data for the system was not easy. But the updating of this data from one semester to the next appears to take only 4 to 8 hours of actual effort, plus some time spent waiting for forms to be returned.

C. Description of the program

1. The interview

The first major item on the advising agenda is the basic interview. The more important portions of the interview are shown flowcharted in Fig. 5. And Fig.6 shows how the program begins. First the student is asked a number of basic items of information: name, student number, age, sex, year in school, degree sought, expected date of graduation, CEEB or GRE scores, and the number of courses the student is interested in taking. He is also asked to indicate how he stands with regard to each course in the departmental list of courses (See Fig. 7). The student is to answer

E (for Equivalently taken) if he has not taken the course for credit but knows the matter anyway or if he is a graduate and got credit for it as an undergraduate (this so the program knows he does not have graduate credit for the course),

A, B, C, D

if he has taken the course and passed with an A, B, C, or D, respectively, except in the case mentioned above,

F if he has taken the course and gotten no credit,

N (for Not taken) if he has neither taken the course nor learned the matter covered,

U (for Unsure) if either E or N is appropriate but he is not sure which,

I (for Information) if he needs more information about the course before answering (this will cause a verbal description to be printed out from ADVISER/CDESCS), or

T (Taking or Transfer) if he expects to have completed the course before the semester for which he is seeking advice or if he has obtained transfer credit for the course.

He is also asked to use E instead of T if he is to become a graduate student as of the beginning of the semester for which he is seeking advice. Problems in wording arose in this section of the interview because of the fact that a student may be using the program a few months before the beginning of the semester for which he is seeking advice or as late as a month or so into the semester itself.

The entire list of topics is then cycled through as the program asks the student about his knowledge and interest in each topic. He is asked to respond to each topic for knowledge

0 if he has no knowledge of it,
1 if he has a little,
2 if he has a fair amount,
3 if he has a good deal, or
4 if he has excellent knowledge of it

and for interest

0 if he has no interest in it,
1 if he has a little,
2 if he has a fair amount,
3 if he has a good deal, or
4 if he has a very high degree of interest in it.

Whenever the student indicates no knowledge of and no interest in a topic, all of its subtopics are skipped and assigned the value 0 for both knowledge and interest.

The program then attempts to guess whether or not the student has equivalently taken any courses to which he has responded with a U (Unsure). First it computes for each course a value called ACHCTKN, where

$$\text{ACHCTKN} = \frac{\displaystyle\sum_{\text{topics}} \left(\begin{matrix}\text{Amount of}\\ \text{coverage}\end{matrix} \times \left(\begin{matrix}\text{Knowledge}\\ \text{level}\end{matrix} - \begin{matrix}\text{Level of}\\ \text{coverage}\end{matrix}\right)\right)}{\text{number of topics covered}}.$$

The higher a course's ACHCTKN is, the more the student knows about the matter covered. Then the average ACHCTKN for all courses taken or equivalently taken or being taken is computed, as is the average ACHCTKN for all courses not taken. The average of these two numbers is then used as a threshold. For each course to which the student responded with a U, the ADVISER will compare its ACHCTKN to this latter average. If ACHCTKN is larger, the program will say, 'I think you know enough of the material in (course designation and title) such that it will probably not benefit you very much to take it. Does this seem reasonable?" It will give the course an E if the student responds YES; otherwise it will give it an N. (YES and NO are the only responses accepted to this particular question.) Corresponding action is taken if ACHCTKN is smaller (See Fig. 8).

At this point the program branches on the basis of the degree sought by the student (Bachelor's in Computer Sciences, Master's in Computer Sciences, Ph.D. in Computer Sciences, none, degree in another department,or Ph.D. minor in Computer Sciences). It will analyze the student's progress toward his degree, telling him exactly which requirements he still has to satisfy (Figs. 9, 10, 11).

The interviewing sections of the program concerned with the course information and with the knowledge and interest digits do, indeed, take a great deal of the student's time (though they probably would not on a display scope, as will be discussed later). Therefore, the student is provided with a short cut option which allows him to input all this information in a very brief fashion, basically with very little output from the program (See Fig. 12). Use of this option does, however, require some pre-preparation of answers by the student. At the end of the basic interview, the program prints out, if the student wants it, an exact copy of what the input and output would look like using the short cut so that he can use the program more quickly should he run it again with the same or very similar data.

Whenever a question is asked of the student, there are only certain answers which are considered sensible by the program, though there are sometimes in reality other sensible answers. Whenever it is not clear what answers are allowed, the program will indicate this to the student. A non-sensible answer will cause the message, "I did not understand your response. Please retype it." to be output. If this happens 10 times in a row, the program quits with an error message. There are a number of cases where the program forces the student to answer YES or NO, though there are other conceivable sensible answers.

2. Weighting and reweighting

All courses are initially given a weight of 0. At the beginning of the weighting section of the program, all courses which have been taken or equivalently taken get 500 subtracted from their weight; if the student is not a graduate student, all graduate level courses get 750 subtracted from their weight; if the student is a Computer Sciences major, certain courses not intended for majors get 438 subtracted from their weight; any courses not currently being offered get 1000 subtracted from their weight. The weights given above serve to eliminate courses from consideration.

Then the following is added to the weight of each course:

$$\frac{\sum_{\text{topics}} \left(\text{Interest} \times \begin{matrix}\text{Amount of}\\ \text{coverage}\end{matrix} \times \left(\begin{matrix}\text{Level of}\\ \text{coverage}\end{matrix} - .75 \times \text{Knowledge}\right)\right)}{\sum_{\text{topics}} \begin{matrix}\text{Amount of}\\ \text{coverage}\end{matrix}}$$

multiplied by a constant which is currently set to 10. The courses are then sorted; and then the top n are printed out, where n represents the number of courses the student expects to take.

This set of courses is then examined to see if there are any problems with it. First the set is examined to see if the courses in it can be taken simultaneously. If they cannot, then a constant, currently 10, is subtracted from a lower-weighted course involved in a conflict. If the courses can be taken without a conflict, then one of the acceptable sets of times, together with the instructors' names will be printed out.

The next thing which the program considers is whether or not the prerequisites of all the courses in the list have been satisfied. For any course for which the prerequisites have not been satisfied, the program will ask the student whether he has obtained the consent of the instructor to take the course. If the answer is YES then the prerequisites will be presumed to be satisfied. If it is NO then the course gets a constant, currently 15, multiplied by the number of courses the student has failed plus 1, subtracted from its weight. The prerequisites which are not satisfied, if they are courses, get the same value, except that the constant is currently 5, added to their weight.

The set of courses is also examined to determine whether or not any two courses are too similar. For each pair of courses in the set, an average difference between the amount of coverage of each topic for the two courses is computed. If the difference is less than 1, the courses are assumed to be very similar, if between 1 and 2 then fairly similar. This similarity is assumed to be a fault in the set of courses if the student is going for a Bachelor's degree and 2 courses are fairly similar or if he is going for a Master's and the courses are very similar. Otherwise, the similarity is assumed to cause no difficulties. In any case, though, the student is asked whether he agrees with the evaluation by either "I imagine that you would not want to take both of these similar courses during the same semester. Do you agree?" or "Though these two courses are similar, I imagine that you would not be opposed on these grounds to taking them both. Is this correct?" The student is thus allowed to disagree with the program and change its evaluation. Whenever the final result of this entire evaluation is that two courses should not be taken together, the lower-weighted of the two courses has its weight decreased by a constant, currently 4.

Next there is a check to determine how much the suggested course list contributes to the student's degree requirements. This procedure is skipped if the student is not working toward a degree or if his degree will be in another department and he is not working toward a Computer Sciences Ph.D. minor. The particulars of this routine are probably not of general interest; so we will discuss only the general principles implemented. If the suggested course list completes the course requirements for the degree sought, then the student is informed of this and the list is assumed to have no deficiencies in this regard. If the set of courses does not complete the degree requirements but all courses in the set contribute to them, the list is again assumed to be all right as regards degree requirements. If the list does not complete the requirements and there are courses in the list which do not contribute, these courses are decreased in weight by a constant, currently 4, multiplied by (the number of courses left to take before requirements are satisfied/the number of courses which it appears can be taken in the time left before the expected date of degree reception) for the Bachelor's or Master's. Only the constant is used in the case of the minor. If the Ph.D. student has not yet taken and passed his screening exam, he is asked whether the suggested course list will help him in his preparation for it. If he answers NO then the program decreases the weight of the top-weighted course by a constant, currently 4. If he has passed his screening exam, then any courses within the department which can contribute to his minor requirements are increased in weight by a constant, currently 4, multiplied by (the number of courses left to be taken before minor requirements are satisfied/the number of courses which can be taken in the time remaining before the expected date of degree reception). Also, if the screening exam has been passed, all courses with positive weight which are concerned with the student's major area (A, B, or C) are increased by a constant, currently 4, multiplied by .5 if Ph.D. qualifiers have been taken and 1 otherwise. Courses concerned with the student's minor area within the department are increased by the constant multiplied by .25 if prelims have been passed and .65 otherwise.

The set of courses as a whole is given a ranking, according to the number and types of difficulties found, as superior, good, fair, poor, very poor, or unacceptable. The student is asked if he agrees with this evaluation (See Fig. 13-15). If he says YES, then the program asks him whether he would rather have it try another list or present a list himself for consideration or stop. If the program is to try another list, it will sort the courses again by weight, look at the new set of top courses, and determine whether this particular set has already been investigated. If it has, the program will reapply all the weights added or subtracted during the last analysis and sort again. If it has done this five times without finding a new set of courses, it will take the highest weighted course not yet fully considered and force it to the top of the list. If it can't find any of these it will ask the student for a list. If it ever turns out that one of the top courses has a negative weight (indicating that it has been eliminated from consideration for some reason), the number of courses to be considered as a set is reduced by one if the student agrees; if there is only one course in the set, the searching for course lists ceases. Whenever the program finds an acceptable-looking new candidate for a course list, it loops back to investigate it for conflicts, similarity, etc., as described above.

If the student indicates that he wishes to present a list of courses for consideration, then an array showing as many of the top-weighted courses as he wants and where they got their weights is printed out. At this point, he is allowed to change the number of courses in the list, if he wishes. He then gives the designations of the courses which he wants considered. If he names a course which has been eliminated, he is told why it was.

If the student disagrees with the evaluation of a set of courses, he is again shown as many of the top-weighted courses and their weights as he wishes. He can tell the program to rate the set of courses as a whole higher or lower. He can also cause courses which he thinks are rated too high to lose weight and courses not included to be placed at the top of the list.

When the program is finished searching for good course lists, either because the student has told it to stop or because it can proceed no further, it then prints out the best five sets of courses encountered, together with any attendant difficulties, in a summary report for the student. Information about current progress toward the student's degree is also included here.

III. Analysis of Results

Eleven students were advised by the program in January, 1968. Of these, 2 were working toward the Bachelor's degree in Computer Sciences, 5 toward the Master's, 2 toward the Ph.D., 1 toward the Ph.D. Minor in Computer Sciences, and 1 toward no degree. The program ended with some questions asking the students to evaluate its performance. Results are shown in Fig. 16. Because of the general agreement among the eleven students and the broad backgrounds represented by them, we feel that the fact that we did run only 11 trials does not seriously detract from the significance of our results.

The length of the basic interview was indeed a problem, as is indicated in the answers to question 5. Of those who used the short cut method of inputting topic knowledge and course information, three said the program took too long, and two said it did not. The basic interview without using short cuts took about 1 1/2 hours; with short cuts it took about 45 minutes. After the basic interview, the student could decide to stop the program at any time. These time considerations are somewhat complicated by the following two facts. (1) Response time was very heavily dependent upon the number and types of jobs currently being handled by the system. (2) The system itself is still experimental and has a tendency to crash.

The program can partially recover from these crashes, but there is still a loss of approximately 15 minutes of real time per crash to the teletype user.

In a production-oriented student advising system, the use of display scopes rather than teletypes might completely solve the time difficulty, since most of the waste results from the fact that the student can read the messages far faster than the teletype can type them. Interviewing techniques would change somewhat with the use of scopes, for the student could no longer look back at something previously typed. But it appears that any difficulties of this sort would be completely outweighted by the saving of real time.

One basic fact seems to stand out when analyzing the answers to questions 1 through 4: Students farther down the road to the Ph.D. were less satisfied, less impressed, and enjoyed the program less than those near the other end of the continuum. Of the two students who did not enjoy using the program, one was a Ph.D. candidate and one a Master's candidate. Of the three students who thought the program inferior to the average human adviser, two were Master's candidates and one was a Ph.D. candidate. Of the three students who thought the program better than the average human adviser, two were Bachelor's candidates and one was not working toward a degree. Nevertheless, only one student was generally dissatisfied. Changes were made in the program to correct the problems he encountered.

(Three or four of the eleven students ran the program before these changes.)

The fact that all eleven students said they would continue to use the program is an indication that our experiment was a success, in spite of the numerous problems encountered. The actual value of the advice given cannot, of course, be judged until later.

One area in which the program was particularly successful was in stating degree requirements at the Bachelor's and Master's levels and telling the student precisely what he had left to do in order to satisfy them. On the other hand, the program was noticeably lacking in the ability to dispense grapevine information.

IV. Generalizability and Cost

The advising program can be run without change each semester. Only the data concerning courses, etc., which is contained on disk, need be changed. An exception to this would occur if the departmental degree requirements changed - in which case pertinent parts of the program would have to be changed accordingly.

Other departments could use the advising program for their students merely by setting up their own course descriptions, prerequisites, etc., except that the program segments dealing with degree requirements would have to be rewritten for the new department (or they could simply be deleted, which would result in some decrease in the quality of the advising).

The economic factor is very difficult to analyze. Runs of the advising program appear to have cost between $10 and $20 of computer time each. This cost is a function of the number of courses, the background of the student, the length of time he keeps the program going, the speed of the machine, the charges per hour, and a number of other factors. And it is very difficult to even guess what effect the use of display scopes instead of teletypes would have on the economic factor.

V. Applications and Implications

It is somewhat difficult to compare this program with similar efforts; for, as far as we know, there have been no other efforts of precisely this nature which have been documented. Some work in the area is going on at SDC (1), Harvard, Penn State, and other places, but we know of no reports on it as yet.

This procedure of computer advising can be likened to information retrieval in a number of ways. We are trying to retrieve for the student a good set of courses, or a number of good sets, from a much larger group of possible sets of courses. The criteria used by the advising program are, however, far more complex than the retrieval criteria of most existing information retrieval systems.

One possible future use of the advising program, or programs like it, might be to combine it with programs like the class scheduling program which we have written (9, 10) and eventually build up a general university service of computer advising and scheduling.

The general advising technique is applicable to many other areas besides that of advising students on courses. We are working with Dr. Warner Slack of the University Medical School on a program to suggest treatments for suspected lung disease, building on some previous work he has done (7, 8). Again we are trying to choose from a large set of alternatives on the basis of rather complex criteria.

Another effort we are currently undertaking involves the generalization of this advising idea. We are trying to set up a program which will write advising programs for human advisers who would rather have the computer do some of their advising. The human adviser will choose the alternatives from which good members are to be sought as well as the criteria for selecting them. This is to be done by means of an interview (via teletype) which will determine precisely how the human adviser wants his advising program to operate. This program-writing program is currently being debugged.

VI. Conclusions

1. The computer can be effectively used to give advice to students concerning which courses they should take.
2. In general, the less advanced the student, the more the computer can do for him.
3. Display scopes would be far superior to teletypes as interviewing devices for student advising.

4. The advice given by the computer was, in certain respects, better than that of a human; and in other respects it was worse.

5. A very useful goal in this area would be the combining of the computer with the human adviser, in such a manner as to utilize the superior qualities of both.

ACKNOWLEDGEMENT

I would like to thank Dr. Leonard Uhr for his encouragement and assistance on this project, the faculty and students of the Computer Sciences Department for the time they spent filling out forms and running the program, and the National Science Foundation for their support through the Graduate Fellowship program. Partial support for this work came from NSF grant GP-7069 and a Wisconsin Alumni Research Foundation grant for computer time.

Bibliography

1. Anonymous, SDC Magazine, Vol. 9, No. 11 (Nov., 1966), pp. 7-8.

2. Anonymous, User Manual Volume IX Revision A UWIS University of Wisconsin Indexing System for Information Handling and Data Indexing, University of Wisconsin Computing Center (Madison, 1968).

3. Anonymous, 3600 3800 Computer Systems INFOL Reference Manual, Control Data Corporation (Palo Alto, 1966).

4. Becker, J., and Hayes, R. M., Information Storage and Retrieval, Wiley (New York, 1963).

5. de Grazia, Alfred, and Sohn, David A. (eds.), Programs, Teachers, and Machines, Bantam Books (New York, 1962).

6. Haefner, Richard R., "Computers and Education," Computer Usage, Vol. 1, No. 3 (Summer, 1966).

7. Slack, W., Peckham, B., Van Cura, L., and Carr, W., "A Computer-Based Physical Examination System," Journal of the American Medical Association, Vol. 200 (Apr. 17, 1967), pp. 224-228.

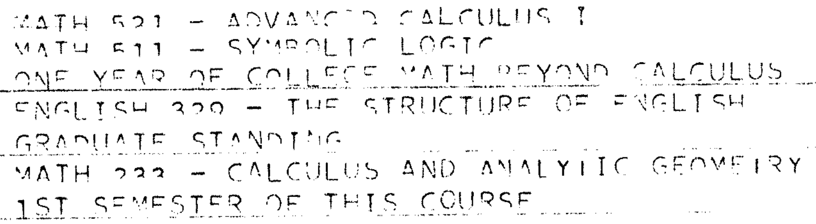
8. Slack, W., Van Cura, L., Hicks, G. P., and Reed, C. E., "A Computer-Based Medical History System," New England Journal of Medicine, Vol. 274 (Jan. 27, 1966), pp. 194-198.

9. Timmreck, E. M., "A User-Oriented Description of SCHEDULER," University of Wisconsin Computer Sciences Department Technical Report No. 2 (Dec., 1966).

10. Timmreck, E. M., "SCHEDULER - A Program That Uses Instructor and Student Preferences to Form Timetables," University of Wisconsin Computer Sciences Department Technical Report No. 3 (June, 1967).

```
IIIA8     CO-OP MONITOR
IIIA9     TOTAL SYSTEMS ENVIRONMENT
IIIB     PROGRAMMING LANGUAGES
IIIB1     PROCEDURE AND PROBLEM ORIENTED LANGUAGES
IIIB1A     FORTRAN
IIIB1B     ALGOL
IIIB1C     COBOL
IIIB1D     PL/I
IIIB1E     COMIT
IIIB1F     JOSS
IIIB1G     TRAC
IIIB2     LIST PROCESSING LANGUAGES
IIIB2A     SLIP
IIIB2B     LISP
IIIB2C     SNOBOL
IIIB2D     REFCO IV
IIIB2E     SAC I
IIIB2F     LIST PROCESSING IN FORTRAN
IIIB2G     LIST PROCESSING IN ALGOL
```

Fig. 1   Some of the topics used in the topic file



```
MATH 521 - ADVANCED CALCULUS I
MATH 511 - SYMBOLIC LOGIC
ONE YEAR OF COLLEGE MATH BEYOND CALCULUS
ENGLISH 329 - THE STRUCTURE OF ENGLISH
GRADUATE STANDING
MATH 233 - CALCULUS AND ANALYTIC GEOMETRY
1ST SEMESTER OF THIS COURSE
```

Fig. 2 A section of the prerequisite file

```
 9TR 1100-1215          13
10TR 1300-1415          15
11TR 1425-1555          12     15
12TR 1500-1630          11     15
13TR 1100-1150           9
```

Fig. 3 Section of time file. Numbers at right indicate overlaps.

```
CS 540INTRODUCTION TO ARTIFICIAL INTELLIGENCE
      3    25
      C     3671121            8                                UHR
11

1           1         1  1             1                                     2

3           11          1 1 1       3132 23 33  32
 1
11

1           1         3  3             1                                     1

2           22         '2 2 1       2122 22 12  32
 1
```

Fig. 4 Section of course description file. Line two contains prerequisites. Line 3 contains area, number of credits, date of last update, time period offered, and instructor. Remaining lines contain "amount of coverage" and "level of coverage" indications.

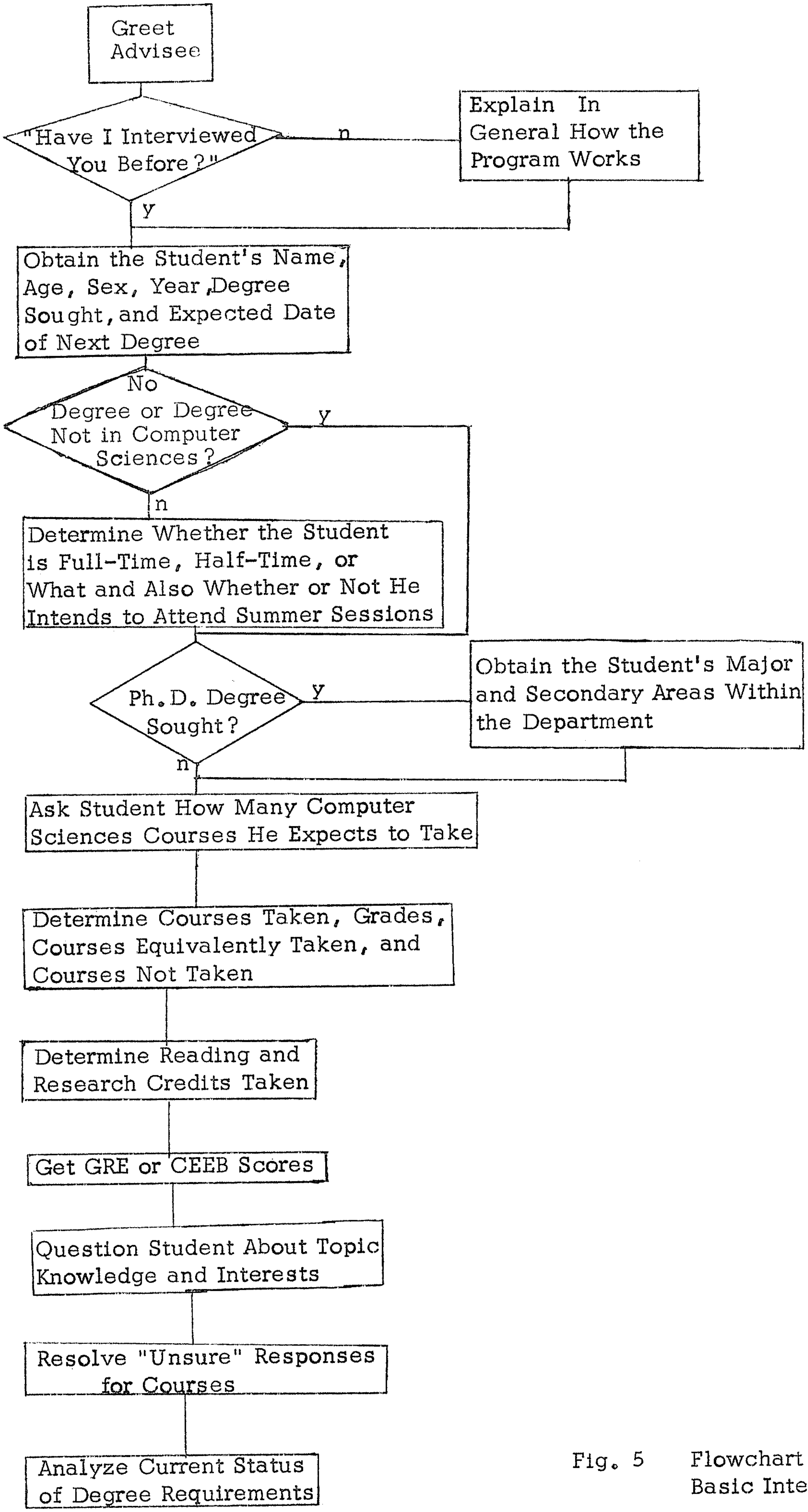
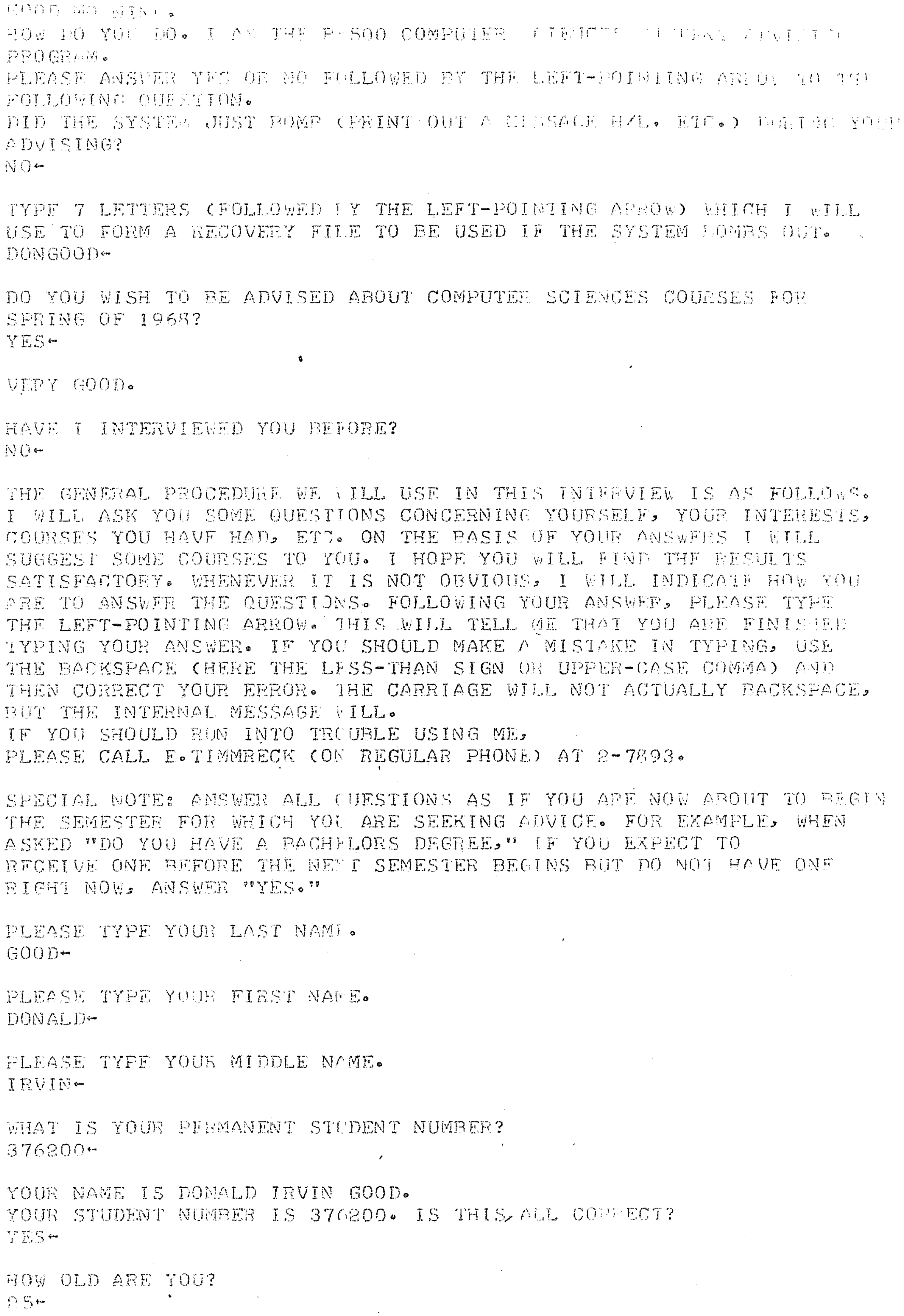
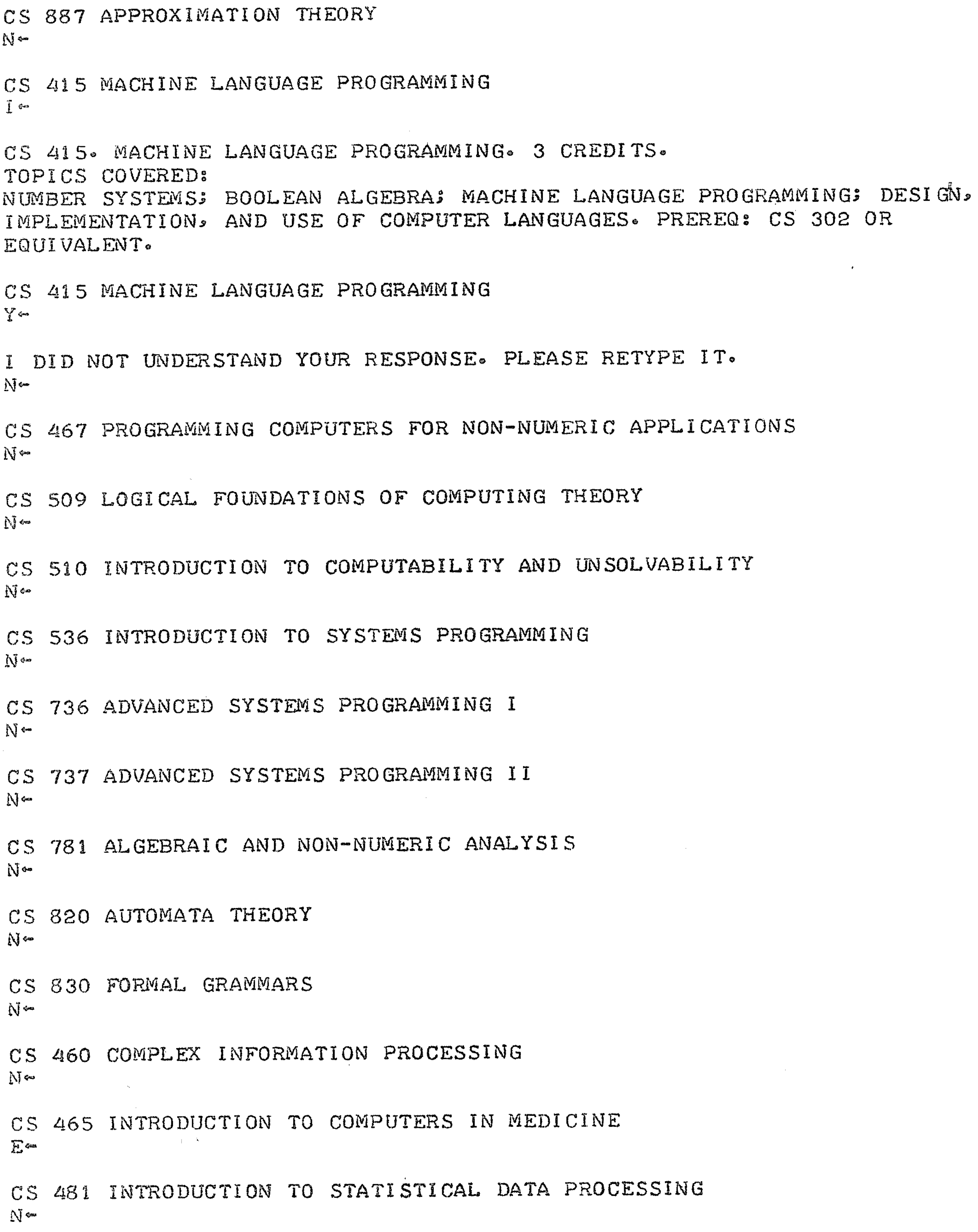

Greet Advisee
"Have I Interviewed You Before?"
n
Explain In General How the Program Works
y
Obtain the Student's Name, Age, Sex, Year, Degree Sought, and Expected Date of Next Degree
No Degree or Degree Not in Computer Sciences?
y
n
Determine Whether the Student is Full-Time, Half-Time, or What and Also Whether or Not He Intends to Attend Summer Sessions
Ph.D. Degree Sought?
y
Obtain the Student's Major and Secondary Areas Within the Department
n
Ask Student How Many Computer Sciences Courses He Expects to Take
Determine Courses Taken, Grades, Courses Equivalently Taken, and Courses Not Taken
Determine Reading and Research Credits Taken
Get GRE or CEEB Scores
Question Student About Topic Knowledge and Interests
Resolve "Unsure" Responses for Courses
Analyze Current Status of Degree Requirements

Fig. 5 Flowchart of Basic Interview



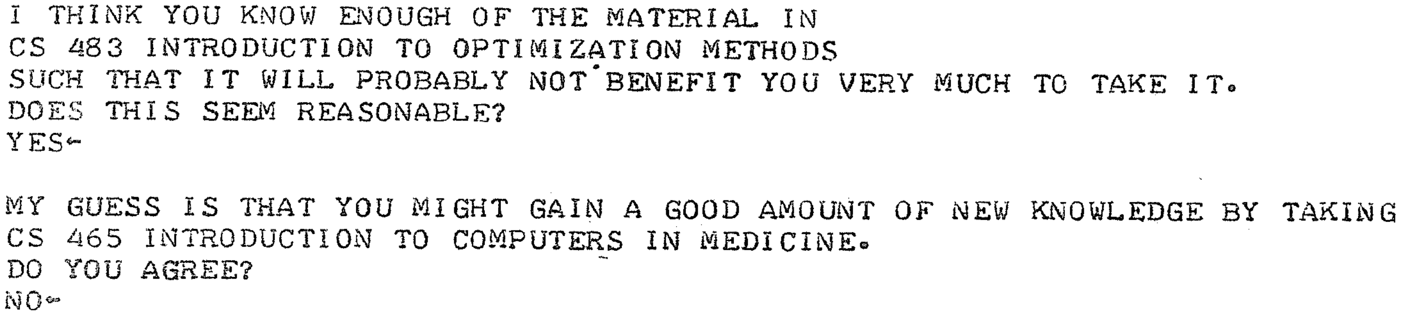
```
GOOD MORNING.
HOW DO YOU DO. I AM THE B-500 COMPUTER DIRECTED STUDENT ADVISING
PROGRAM.
PLEASE ANSWER YES OR NO FOLLOWED BY THE LEFT-POINTING ARROW TO THE
FOLLOWING QUESTION.
DID THE SYSTEM JUST BOMB (PRINT OUT A MESSAGE H/L, ETC.) DURING YOUR
ADVISING?
NO←

TYPE 7 LETTERS (FOLLOWED BY THE LEFT-POINTING ARROW) WHICH I WILL
USE TO FORM A RECOVERY FILE TO BE USED IF THE SYSTEM BOMBS OUT.
DONGOOD←

DO YOU WISH TO BE ADVISED ABOUT COMPUTER SCIENCES COURSES FOR
SPRING OF 1968?
YES←

VERY GOOD.

HAVE I INTERVIEWED YOU BEFORE?
NO←

THE GENERAL PROCEDURE WE WILL USE IN THIS INTERVIEW IS AS FOLLOWS.
I WILL ASK YOU SOME QUESTIONS CONCERNING YOURSELF, YOUR INTERESTS,
COURSES YOU HAVE HAD, ETC. ON THE BASIS OF YOUR ANSWERS I WILL
SUGGEST SOME COURSES TO YOU. I HOPE YOU WILL FIND THE RESULTS
SATISFACTORY. WHENEVER IT IS NOT OBVIOUS, I WILL INDICATE HOW YOU
ARE TO ANSWER THE QUESTIONS. FOLLOWING YOUR ANSWER, PLEASE TYPE
THE LEFT-POINTING ARROW. THIS WILL TELL ME THAT YOU ARE FINISHED
TYPING YOUR ANSWER. IF YOU SHOULD MAKE A MISTAKE IN TYPING, USE
THE BACKSPACE (HERE THE LESS-THAN SIGN OR UPPER-CASE COMMA) AND
THEN CORRECT YOUR ERROR. THE CARRIAGE WILL NOT ACTUALLY BACKSPACE,
BUT THE INTERNAL MESSAGE WILL.
IF YOU SHOULD RUN INTO TROUBLE USING ME,
PLEASE CALL E.TIMMRECK (ON REGULAR PHONE) AT 2-7893.

SPECIAL NOTE: ANSWER ALL QUESTIONS AS IF YOU ARE NOW ABOUT TO BEGIN
THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE. FOR EXAMPLE, WHEN
ASKED "DO YOU HAVE A BACHELORS DEGREE," IF YOU EXPECT TO
RECEIVE ONE BEFORE THE NEXT SEMESTER BEGINS BUT DO NOT HAVE ONE
RIGHT NOW, ANSWER "YES."

PLEASE TYPE YOUR LAST NAME.
GOOD←

PLEASE TYPE YOUR FIRST NAME.
DONALD←

PLEASE TYPE YOUR MIDDLE NAME.
IRVIN←

WHAT IS YOUR PERMANENT STUDENT NUMBER?
376200←

YOUR NAME IS DONALD IRVIN GOOD.
YOUR STUDENT NUMBER IS 376200. IS THIS ALL CORRECT?
YES←

HOW OLD ARE YOU?
25←
```

Fig. 6 Portion of basic interview



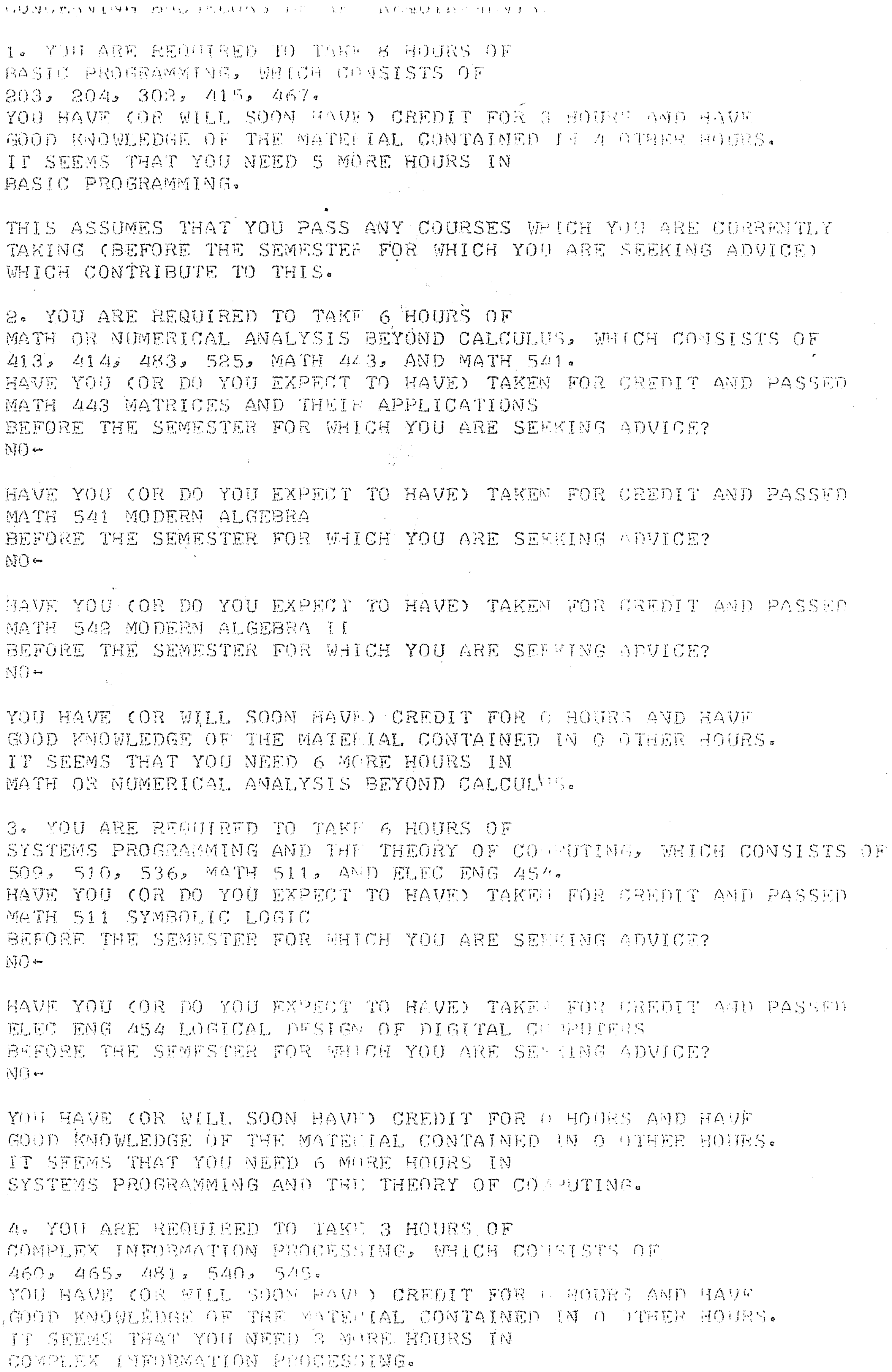
```
CS 887 APPROXIMATION THEORY
N←

CS 415 MACHINE LANGUAGE PROGRAMMING
I←

CS 415. MACHINE LANGUAGE PROGRAMMING. 3 CREDITS.
TOPICS COVERED:
NUMBER SYSTEMS; BOOLEAN ALGEBRA; MACHINE LANGUAGE PROGRAMMING; DESIGN,
IMPLEMENTATION, AND USE OF COMPUTER LANGUAGES. PREREQ: CS 302 OR
EQUIVALENT.

CS 415 MACHINE LANGUAGE PROGRAMMING
Y←

I DID NOT UNDERSTAND YOUR RESPONSE. PLEASE RETYPE IT.
N←

CS 467 PROGRAMMING COMPUTERS FOR NON-NUMERIC APPLICATIONS
N←

CS 509 LOGICAL FOUNDATIONS OF COMPUTING THEORY
N←

CS 510 INTRODUCTION TO COMPUTABILITY AND UNSOLVABILITY
N←

CS 536 INTRODUCTION TO SYSTEMS PROGRAMMING
N←

CS 736 ADVANCED SYSTEMS PROGRAMMING I
N←

CS 737 ADVANCED SYSTEMS PROGRAMMING II
N←

CS 781 ALGEBRAIC AND NON-NUMERIC ANALYSIS
N←

CS 820 AUTOMATA THEORY
N←

CS 830 FORMAL GRAMMARS
N←

CS 460 COMPLEX INFORMATION PROCESSING
N←

CS 465 INTRODUCTION TO COMPUTERS IN MEDICINE
E←

CS 481 INTRODUCTION TO STATISTICAL DATA PROCESSING
N←
```

Fig. 7   Questioning the student about courses



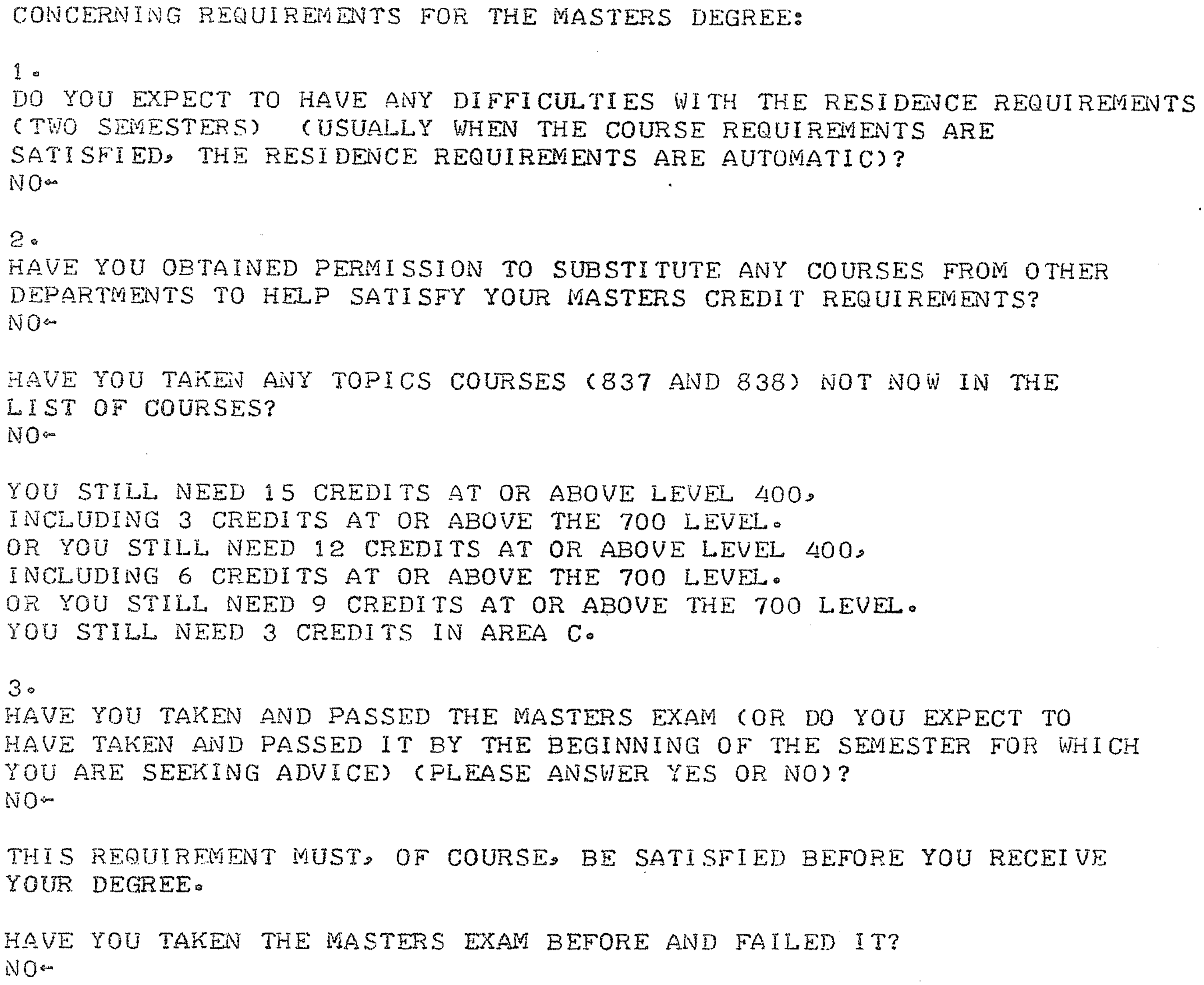
```
I THINK YOU KNOW ENOUGH OF THE MATERIAL IN
CS 483 INTRODUCTION TO OPTIMIZATION METHODS
SUCH THAT IT WILL PROBABLY NOT BENEFIT YOU VERY MUCH TO TAKE IT.
DOES THIS SEEM REASONABLE?
YES←

MY GUESS IS THAT YOU MIGHT GAIN A GOOD AMOUNT OF NEW KNOWLEDGE BY TAKING
CS 465 INTRODUCTION TO COMPUTERS IN MEDICINE.
DO YOU AGREE?
NO←
```

Fig. 8 Resolving the "UNSURE" response for courses



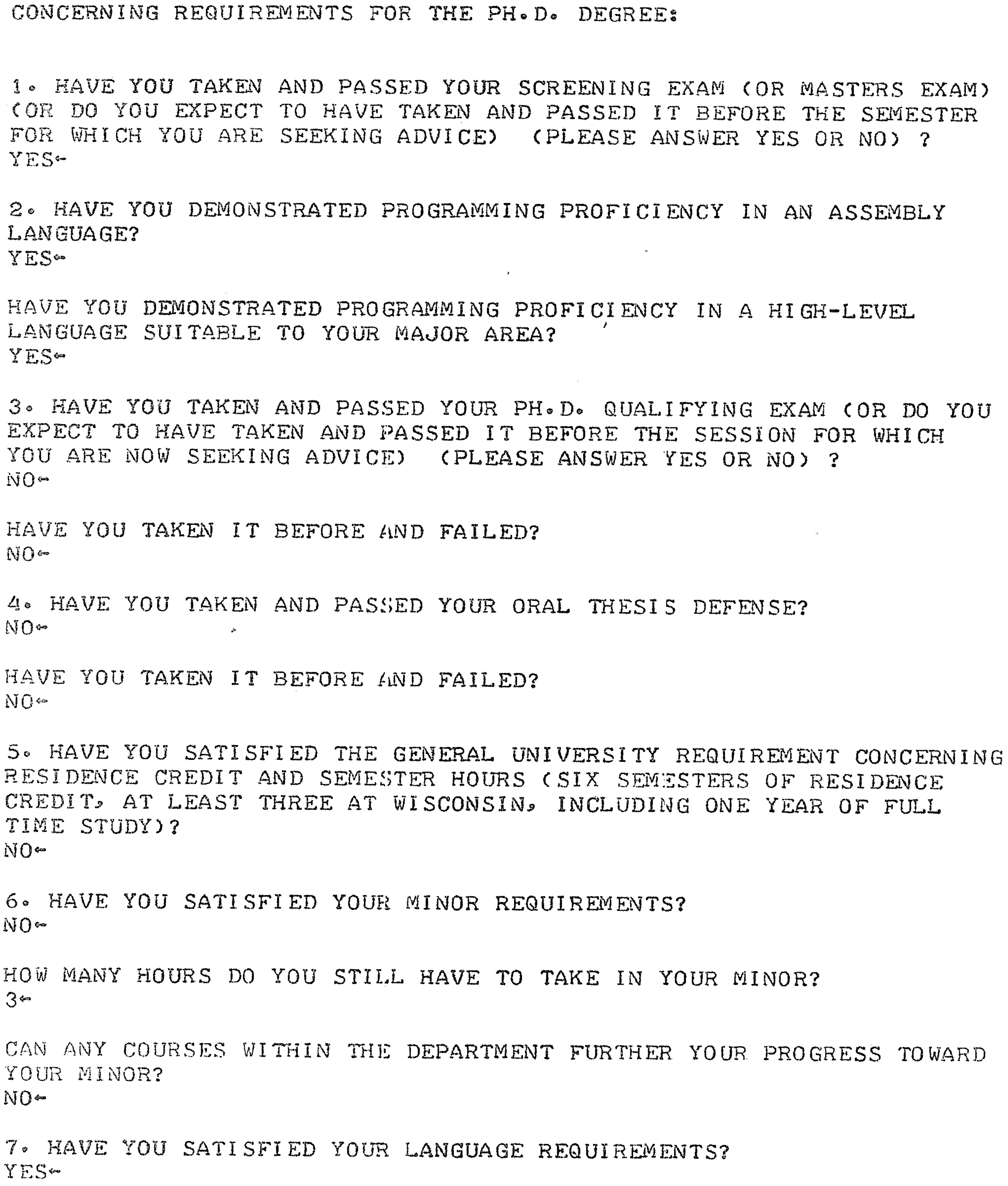
```
CONCERNING PROGRESS [illegible] REQUIREMENTS

1. YOU ARE REQUIRED TO TAKE 8 HOURS OF
BASIC PROGRAMMING, WHICH CONSISTS OF
203, 204, 302, 415, 467.
YOU HAVE (OR WILL SOON HAVE) CREDIT FOR 3 HOURS AND HAVE
GOOD KNOWLEDGE OF THE MATERIAL CONTAINED IN 4 OTHER HOURS.
IT SEEMS THAT YOU NEED 5 MORE HOURS IN
BASIC PROGRAMMING.

THIS ASSUMES THAT YOU PASS ANY COURSES WHICH YOU ARE CURRENTLY
TAKING (BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE)
WHICH CONTRIBUTE TO THIS.

2. YOU ARE REQUIRED TO TAKE 6 HOURS OF
MATH OR NUMERICAL ANALYSIS BEYOND CALCULUS, WHICH CONSISTS OF
413, 414, 483, 525, MATH 443, AND MATH 541.
HAVE YOU (OR DO YOU EXPECT TO HAVE) TAKEN FOR CREDIT AND PASSED
MATH 443 MATRICES AND THEIR APPLICATIONS
BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE?
NO←

HAVE YOU (OR DO YOU EXPECT TO HAVE) TAKEN FOR CREDIT AND PASSED
MATH 541 MODERN ALGEBRA
BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE?
NO←

HAVE YOU (OR DO YOU EXPECT TO HAVE) TAKEN FOR CREDIT AND PASSED
MATH 542 MODERN ALGEBRA II
BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE?
NO←

YOU HAVE (OR WILL SOON HAVE) CREDIT FOR 0 HOURS AND HAVE
GOOD KNOWLEDGE OF THE MATERIAL CONTAINED IN 0 OTHER HOURS.
IT SEEMS THAT YOU NEED 6 MORE HOURS IN
MATH OR NUMERICAL ANALYSIS BEYOND CALCULUS.

3. YOU ARE REQUIRED TO TAKE 6 HOURS OF
SYSTEMS PROGRAMMING AND THE THEORY OF COMPUTING, WHICH CONSISTS OF
509, 510, 536, MATH 511, AND ELEC ENG 454.
HAVE YOU (OR DO YOU EXPECT TO HAVE) TAKEN FOR CREDIT AND PASSED
MATH 511 SYMBOLIC LOGIC
BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE?
NO←

HAVE YOU (OR DO YOU EXPECT TO HAVE) TAKEN FOR CREDIT AND PASSED
ELEC ENG 454 LOGICAL DESIGN OF DIGITAL COMPUTERS
BEFORE THE SEMESTER FOR WHICH YOU ARE SEEKING ADVICE?
NO←

YOU HAVE (OR WILL SOON HAVE) CREDIT FOR 0 HOURS AND HAVE
GOOD KNOWLEDGE OF THE MATERIAL CONTAINED IN 0 OTHER HOURS.
IT SEEMS THAT YOU NEED 6 MORE HOURS IN
SYSTEMS PROGRAMMING AND THE THEORY OF COMPUTING.

4. YOU ARE REQUIRED TO TAKE 3 HOURS OF
COMPLEX INFORMATION PROCESSING, WHICH CONSISTS OF
460, 465, 481, 540, 545.
YOU HAVE (OR WILL SOON HAVE) CREDIT FOR 0 HOURS AND HAVE
GOOD KNOWLEDGE OF THE MATERIAL CONTAINED IN 0 OTHER HOURS.
IT SEEMS THAT YOU NEED 3 MORE HOURS IN
COMPLEX INFORMATION PROCESSING.
```

Fig. 9 Partial analysis of Bachelor's requirements



```
CONCERNING REQUIREMENTS FOR THE MASTERS DEGREE:

1.
DO YOU EXPECT TO HAVE ANY DIFFICULTIES WITH THE RESIDENCE REQUIREMENTS
(TWO SEMESTERS)  (USUALLY WHEN THE COURSE REQUIREMENTS ARE
SATISFIED, THE RESIDENCE REQUIREMENTS ARE AUTOMATIC)?
NO←

2.
HAVE YOU OBTAINED PERMISSION TO SUBSTITUTE ANY COURSES FROM OTHER
DEPARTMENTS TO HELP SATISFY YOUR MASTERS CREDIT REQUIREMENTS?
NO←

HAVE YOU TAKEN ANY TOPICS COURSES (837 AND 838) NOT NOW IN THE
LIST OF COURSES?
NO←

YOU STILL NEED 15 CREDITS AT OR ABOVE LEVEL 400,
INCLUDING 3 CREDITS AT OR ABOVE THE 700 LEVEL.
OR YOU STILL NEED 12 CREDITS AT OR ABOVE LEVEL 400,
INCLUDING 6 CREDITS AT OR ABOVE THE 700 LEVEL.
OR YOU STILL NEED 9 CREDITS AT OR ABOVE THE 700 LEVEL.
YOU STILL NEED 3 CREDITS IN AREA C.

3.
HAVE YOU TAKEN AND PASSED THE MASTERS EXAM (OR DO YOU EXPECT TO
HAVE TAKEN AND PASSED IT BY THE BEGINNING OF THE SEMESTER FOR WHICH
YOU ARE SEEKING ADVICE) (PLEASE ANSWER YES OR NO)?
NO←

THIS REQUIREMENT MUST, OF COURSE, BE SATISFIED BEFORE YOU RECEIVE
YOUR DEGREE.

HAVE YOU TAKEN THE MASTERS EXAM BEFORE AND FAILED IT?
NO←
```

Fig. 10   Analysis of Master's requirements

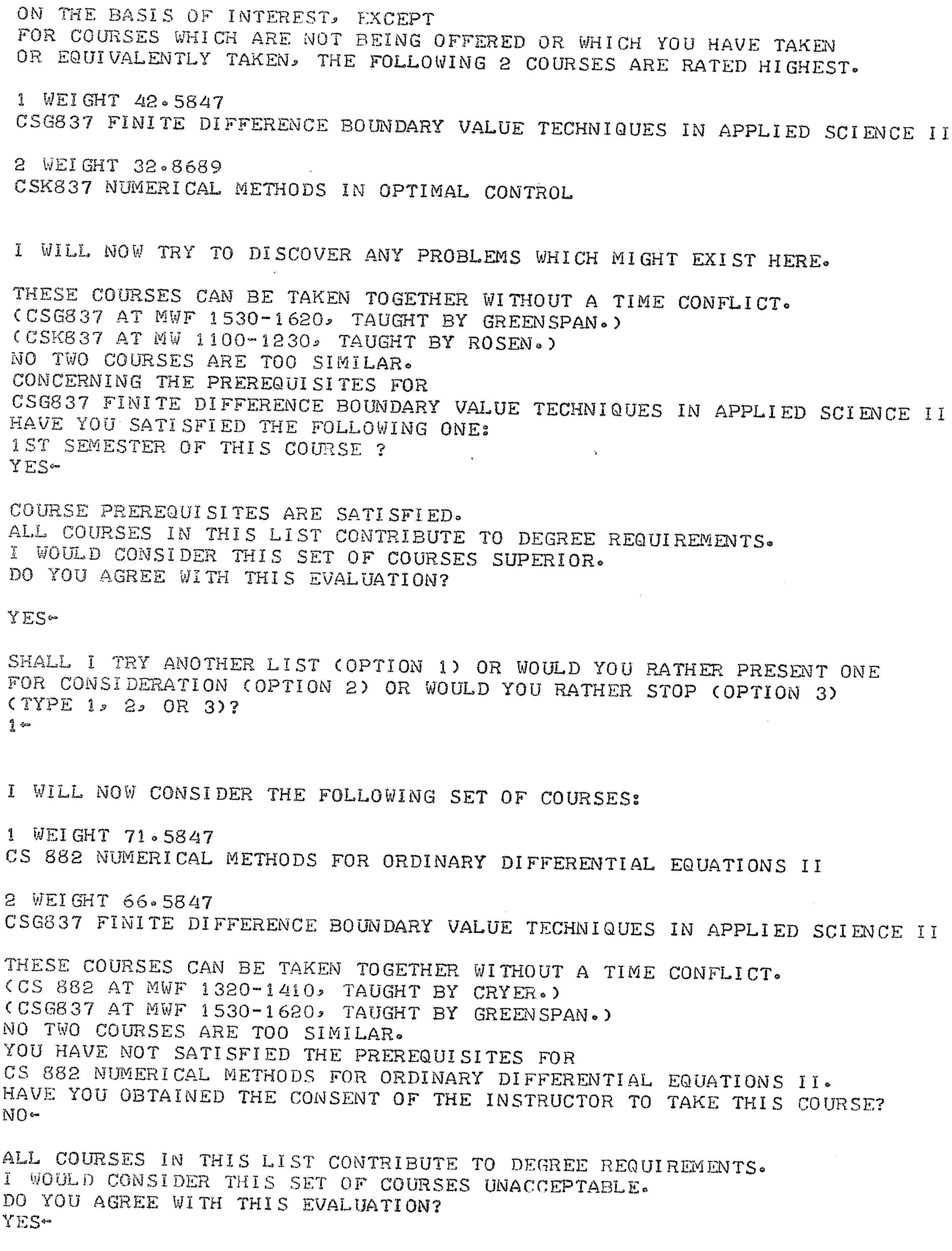


```
CONCERNING REQUIREMENTS FOR THE PH.D. DEGREE:

1. HAVE YOU TAKEN AND PASSED YOUR SCREENING EXAM (OR MASTERS EXAM)
(OR DO YOU EXPECT TO HAVE TAKEN AND PASSED IT BEFORE THE SEMESTER
FOR WHICH YOU ARE SEEKING ADVICE)  (PLEASE ANSWER YES OR NO) ?
YES←

2. HAVE YOU DEMONSTRATED PROGRAMMING PROFICIENCY IN AN ASSEMBLY
LANGUAGE?
YES←

HAVE YOU DEMONSTRATED PROGRAMMING PROFICIENCY IN A HIGH-LEVEL
LANGUAGE SUITABLE TO YOUR MAJOR AREA?
YES←

3. HAVE YOU TAKEN AND PASSED YOUR PH.D. QUALIFYING EXAM (OR DO YOU
EXPECT TO HAVE TAKEN AND PASSED IT BEFORE THE SESSION FOR WHICH
YOU ARE NOW SEEKING ADVICE)  (PLEASE ANSWER YES OR NO) ?
NO←

HAVE YOU TAKEN IT BEFORE AND FAILED?
NO←

4. HAVE YOU TAKEN AND PASSED YOUR ORAL THESIS DEFENSE?
NO←

HAVE YOU TAKEN IT BEFORE AND FAILED?
NO←

5. HAVE YOU SATISFIED THE GENERAL UNIVERSITY REQUIREMENT CONCERNING
RESIDENCE CREDIT AND SEMESTER HOURS (SIX SEMESTERS OF RESIDENCE
CREDIT, AT LEAST THREE AT WISCONSIN, INCLUDING ONE YEAR OF FULL
TIME STUDY)?
NO←

6. HAVE YOU SATISFIED YOUR MINOR REQUIREMENTS?
NO←

HOW MANY HOURS DO YOU STILL HAVE TO TAKE IN YOUR MINOR?
3←

CAN ANY COURSES WITHIN THE DEPARTMENT FURTHER YOUR PROGRESS TOWARD
YOUR MINOR?
NO←

7. HAVE YOU SATISFIED YOUR LANGUAGE REQUIREMENTS?
YES←
```

Fig. 11   Analysis of Ph.D. requirements

```
DO YOU BOTH KNOW HOW TO AND WANT TO USE THE QUICK METHOD OF
INPUTTING TOPIC KNOWLEDGE AND INTEREST?
YES←

KNOWLEDGE
16 TOPICS...
2111222112111111←

72 TOPICS...
332333333332223223333322322333231332233323323000012210001000000000000008

23 TOPICS...
33332133200000111111100←

72 TOPICS...
333332232333340100002222202003300000102222111000111111111000000000000004

11 TOPICS...
32111111000←

34 TOPICS...
1100120201000000000000000000000000←

72 TOPICS...
111122221110000000000000000000000000000000000000000000000000000000000008

 4 TOPICS...
0<0000←

INTEREST
16 TOPICS...
2111222133222212←

72 TOPICS...
222222222222111221222122222222222122212221121222222222211221112111221212128

23 TOPICS...
22211133122222222221111←

72 TOPICS...
333332232233333210002222222221330021220333323331111111111113113222222222228

11 TOPICS...
33222232211←

34 TOPICS...
2111122222222222222333222333333222←

72 TOPICS...
3223222222331111100000011121211113333332321222311111111002121111110000114

 4 TOPICS...
1122←

LET ME THINK FOR A MINUTE OR TWO.
```

Fig. 12 The short cut method of topic input. Each digit refers to a particular topic.



```
ON THE BASIS OF INTEREST, EXCEPT
FOR COURSES WHICH ARE NOT BEING OFFERED OR WHICH YOU HAVE TAKEN
OR EQUIVALENTLY TAKEN, THE FOLLOWING 2 COURSES ARE RATED HIGHEST.

1 WEIGHT 42.5847
CSG837 FINITE DIFFERENCE BOUNDARY VALUE TECHNIQUES IN APPLIED SCIENCE II

2 WEIGHT 32.8689
CSK837 NUMERICAL METHODS IN OPTIMAL CONTROL


I WILL NOW TRY TO DISCOVER ANY PROBLEMS WHICH MIGHT EXIST HERE.

THESE COURSES CAN BE TAKEN TOGETHER WITHOUT A TIME CONFLICT.
(CSG837 AT MWF 1530-1620, TAUGHT BY GREENSPAN.)
(CSK837 AT MW 1100-1230, TAUGHT BY ROSEN.)
NO TWO COURSES ARE TOO SIMILAR.
CONCERNING THE PREREQUISITES FOR
CSG837 FINITE DIFFERENCE BOUNDARY VALUE TECHNIQUES IN APPLIED SCIENCE II
HAVE YOU SATISFIED THE FOLLOWING ONE:
1ST SEMESTER OF THIS COURSE ?
YES←

COURSE PREREQUISITES ARE SATISFIED.
ALL COURSES IN THIS LIST CONTRIBUTE TO DEGREE REQUIREMENTS.
I WOULD CONSIDER THIS SET OF COURSES SUPERIOR.
DO YOU AGREE WITH THIS EVALUATION?

YES←

SHALL I TRY ANOTHER LIST (OPTION 1) OR WOULD YOU RATHER PRESENT ONE
FOR CONSIDERATION (OPTION 2) OR WOULD YOU RATHER STOP (OPTION 3)
(TYPE 1, 2, OR 3)?
1←


I WILL NOW CONSIDER THE FOLLOWING SET OF COURSES:

1 WEIGHT 71.5847
CS 882 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS II

2 WEIGHT 66.5847
CSG837 FINITE DIFFERENCE BOUNDARY VALUE TECHNIQUES IN APPLIED SCIENCE II

THESE COURSES CAN BE TAKEN TOGETHER WITHOUT A TIME CONFLICT.
(CS 882 AT MWF 1320-1410, TAUGHT BY CRYER.)
(CSG837 AT MWF 1530-1620, TAUGHT BY GREENSPAN.)
NO TWO COURSES ARE TOO SIMILAR.
YOU HAVE NOT SATISFIED THE PREREQUISITES FOR
CS 882 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS II.
HAVE YOU OBTAINED THE CONSENT OF THE INSTRUCTOR TO TAKE THIS COURSE?
NO←

ALL COURSES IN THIS LIST CONTRIBUTE TO DEGREE REQUIREMENTS.
I WOULD CONSIDER THIS SET OF COURSES UNACCEPTABLE.
DO YOU AGREE WITH THIS EVALUATION?
YES←
```

Fig. 13 Evaluating sets of courses

```
I WILL NOW CONSIDER THE FOLLOWING SET OF COURSES:

1 WEIGHT 43.9815
CS 536 INTRODUCTION TO SYSTEMS PROGRAMMING

2 WEIGHT 30.5357
CS 510 INTRODUCTION TO COMPUTABILITY AND UNSOLVABILITY

3 WEIGHT 30.0581
CS 540 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

4 WEIGHT 25.9211
CS 509 LOGICAL FOUNDATIONS OF COMPUTING THEORY

THESE COURSES CAN BE TAKEN TOGETHER WITHOUT A TIME CONFLICT.
(CS 536 AT TR 1425-1555, TAUGHT BY WILLIAMS.)
(CS 510 AT TR 1300-1415, TAUGHT BY CUDIA.)
(CS 540 AT TR 0850-1030, TAUGHT BY UHR.)
(CS 509 AT MWF 0850-0940, TAUGHT BY LANDWEBER.)
NO TWO COURSES ARE TOO SIMILAR.
CONCERNING THE PREREQUISITES FOR
CS 510 INTRODUCTION TO COMPUTABILITY AND UNSOLVABILITY
HAVE YOU SATISFIED THE FOLLOWING ONE:
MATH 511 - SYMBOLIC LOGIC ?
NO←

CONCERNING THE PREREQUISITES FOR
CS 510 INTRODUCTION TO COMPUTABILITY AND UNSOLVABILITY
HAVE YOU SATISFIED THE FOLLOWING ONE:
ONE YEAR OF COLLEGE MATH BEYOND CALCULUS ?
YES←

COURSE PREREQUISITES ARE SATISFIED.
I WOULD CONSIDER THIS SET OF COURSES SUPERIOR.
DO YOU AGREE WITH THIS EVALUATION?
NO←

I HAVE AN ARRAY WHICH SHOWS THE COURSES, THEIR WEIGHTS, AND HOW
THEY GOT THEM. "INT" STANDS FOR INTEREST. "-PREREQ"
INDICATES WEIGHTS LOST BY COURSES NEEDING PREREQUISITES.
"+PREREQ" INDICATES WEIGHTS ADDED TO NEEDED PREREQUISITES.
"SIM" STANDS FOR WEIGHTS LOST BECAUSE TWO COURSES WERE TOO
SIMILAR. "DEGREE" STANDS FOR WEIGHTS RESULTING FROM DEGREE
REQUIREMENTS. "CON" STANDS FOR WEIGHTS RESULTING FROM CONFLICTS
BETWEEN COURSES. "OTHER" REFERS TO WEIGHTS ASSIGNED BECAUSE
COURSE IS NOT BEING OFFERED, WILL NOT HELP YOU, OR HAS BEEN FORCED
HIGHER OR LOWER BY YOU.
FOR HOW MANY OF THE TOP WEIGHTED COURSES WOULD YOU LIKE TO SEE THE
CURRENT WEIGHT AND WHERE IT CAME FROM?
10←

COURSE     TOTAL    INT    -PREREQ  +PREREQ   SIM    DEGREE    CON     OTHER
CS 536     43.98   28.98     0.00    15.00    0.00    0.00     0.00    0.00
CS 510     30.54   25.54     0.00     5.00    0.00    0.00     0.00    0.00
CS 540     30.06   30.06     0.00     0.00    0.00    0.00     0.00    0.00
CS 509     25.92   25.92     0.00     0.00    0.00    0.00     0.00    0.00
CSL838     22.79   37.79   -15.00     0.00    0.00    0.00     0.00    0.00
CSK838     22.33   47.33   -15.00     0.00    0.00    0.00   -10.00    0.00
CS 545     21.25   36.25   -15.00     0.00    0.00    0.00     0.00    0.00
CSM838     21.18   36.18   -15.00     0.00    0.00    0.00     0.00    0.00
CS 467     20.31   20.31     0.00     0.00    0.00    0.00     0.00    0.00
CS 830     19.46   49.46   -30.00     0.00    0.00    0.00     0.00    0.00
```

```
DO YOU THINK THAT THIS LAST SUGGESTED SET OF COURSES AS A WHOLE IS
BETTER THAN I HAVE CALCULATED IT TO BE?
NO←

DO YOU THINK THAT IT IS WORSE?
NO←

DO YOU THINK THAT ANY COURSES IN THIS LIST ARE RATED TOO HIGH?
YES←

HOW MANY COURSES ARE RATED TOO HIGH?
1←

IS CS 536 RATED TOO HIGH?
NO←

IS CS 510 RATED TOO HIGH?
YES←

WOULD YOU LIKE TO SPECIFY ANY OTHER COURSE OR COURSES TO BE PLACED
AT THE TOP OF THE LIST?
YES←

GIVE THE DESIGNATION (CS XXX) OF ONE SUCH COURSE.
CS545←

ARE THERE ANY MORE COURSES IN THIS CATEGORY?
YES←

GIVE THE DESIGNATION OF ONE.
CS741←

ARE THERE ANY MORE COURSES IN THIS CATEGORY?
NO←


I WILL NOW CONSIDER THE FOLLOWING SET OF COURSES:

1 WEIGHT 48.9815
CS 545 NATURAL LANGUAGE AND COMPUTING

2 WEIGHT 48.9815
CS 741 INFORMATION RETRIEVAL THEORY AND TECHNIQUES

3 WEIGHT 43.9815
CS 536 INTRODUCTION TO SYSTEMS PROGRAMMING

4 WEIGHT 30.0581
CS 540 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

THESE COURSES CAN BE TAKEN TOGETHER WITHOUT A TIME CONFLICT.
(CS 545 AT MWF 1100-1150, TAUGHT BY VENEZKY.)
(CS 741 AT MWF 1320-1410, TAUGHT BY WYLLYS.)
(CS 536 AT TR 1425-1555, TAUGHT BY WILLIAMS.)
(CS 540 AT TR 0850-1030, TAUGHT BY UHR.)
NO TWO COURSES ARE TOO SIMILAR.
YOU HAVE NOT SATISFIED THE PREREQUISITES FOR CS 545.
I WOULD CONSIDER THIS SET OF COURSES UNACCEPTABLE.
DO YOU AGREE WITH THIS EVALUATION?
NO←

PRINT WEIGHT ARRAY FOR HOW MANY COURSES?
0←
```

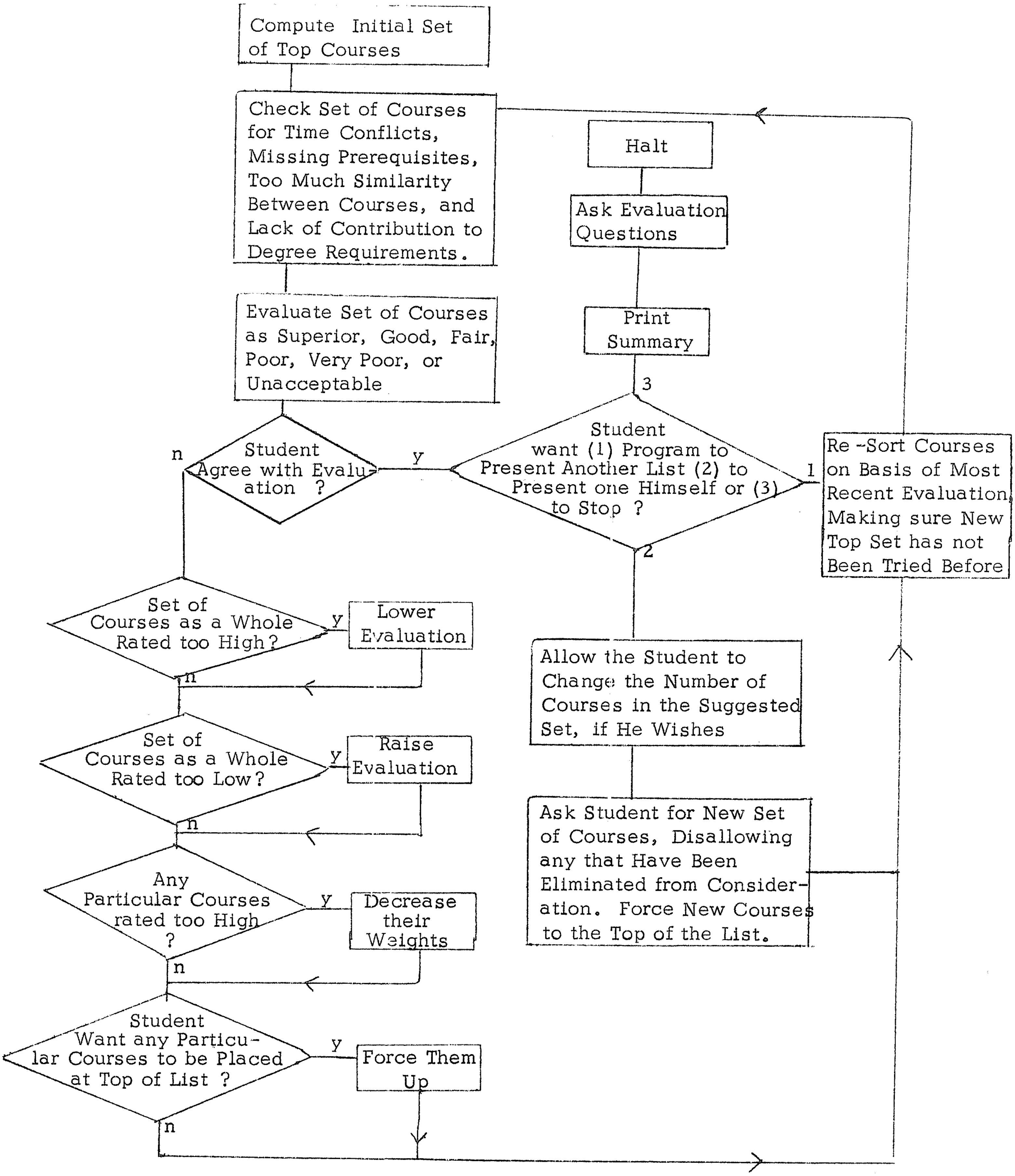

Compute Initial Set of Top Courses
Check Set of Courses for Time Conflicts, Missing Prerequisites, Too Much Similarity Between Courses, and Lack of Contribution to Degree Requirements.
Evaluate Set of Courses as Superior, Good, Fair, Poor, Very Poor, or Unacceptable
Student Agree with Evaluation ?
n
y
Halt
Ask Evaluation Questions
Print Summary
3
Student want (1) Program to Present Another List (2) to Present one Himself or (3) to Stop ?
1
2
Re -Sort Courses on Basis of Most Recent Evaluation, Making sure New Top Set has not Been Tried Before
Set of Courses as a Whole Rated too High?
y
Lower Evaluation
n
Set of Courses as a Whole Rated too Low?
y
Raise Evaluation
n
Allow the Student to Change the Number of Courses in the Suggested Set, if He Wishes
Ask Student for New Set of Courses, Disallowing any that Have Been Eliminated from Consideration. Force New Courses to the Top of the List.
Any Particular Courses rated too High ?
y
Decrease their Weights
n
Student Want any Particular Courses to be Placed at Top of List ?
y
Force Them Up
n

Fig. 15 Flowchart of searching procedure with interaction

		YES	NO
1.	Are you in general satisfied with the results of this program?	10	1
2.	Did you enjoy using the program?	9	2
3.	Do you think the program did as well as the average human adviser?	8	3
4.	Do you think the program did better than the average human adviser?	3	8
5.	Do you think the program took too long?	8	3
6.	Do you think this advising procedure can be shortened without losing accuracy?	5	6
7.	If this program is maintained and properly updated for use each semester, will you continue to use it?	11	0

Fig. 16 Evaluation questions and results