AN AUTOMATED SYSTEM FOR UNIVERSITY PLACEMENT CENTER
INTERVIEW SCHEDULING AND DATABASE MANAGEMENT

BY

JOHN C. DEBO
B.S., Iowa State University, 1963
M.Ed., Florida Atlantic University, 1982

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program
of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1985
COPYRIGHT © 1985
By John C. Debo
ABSTRACT

As the University of Central Florida grows, the number of students using the placement center increases and the number of prospective employers conducting interviews also increases rapidly. The present system for scheduling interviews and maintaining records on students and employers is time and paperwork intensive, and is falling behind the load.

The system described in this paper is designed to do most of the interaction with students seeking interviews to aid the staff in scheduling the employers' interviewing days, and to do most of the record keeping of eligible students and prospective employers.

The several program modules are written in Basic language on the IBM Personal Computer in order that: 1) They may be easily updated, trouble shot, and modified by student assistants. 2) An individual task may be revised without affecting all the other tasks which the system does. 3) The system be as "user friendly" as possible, thus not requiring computer literacy on the part of staff and students. 4) The system would be kept completely under the control of the placement center staff so that it can be made available on whatever schedule is deemed best; this also would enhance security of the database.
ACKNOWLEDGEMENTS

In memory of my mother, Virginia Carpenter Debo.
I wish to thank my committee chairman, Dr. Christian S. Bauer, who is one of the most humane beings in the College of Engineering.

Thanks to Dr. Richard G. Denning, who has quite literally made the entire project possible for me.

Thanks to Dr. Flora Pinder, who suggested the need for the project and has followed it to completion.

Finally, thanks to my father, Lewis Debo, who has helped all through the final weeks of this project with everything from editorial assistance to being a good cook.
PREFACE

About twice a month you will find students bedding down in sleeping bags outside the Administration Building, in order to be in line when placement interview scheduling opens the next morning.

A great deal of staff time is used in interacting with the students when they do sign up for interviews the next day, as well as in the paperwork required to manage the student files, the interviewer schedules, and the files of upcoming interviews.

A great deal of time is consumed in writing out the interviewer's schedule of appointments for the day and in writing letters to students who miss their scheduled interview.

This situation led Dr. Flora Pinder, formerly with the University Placement Center, to suggest to me that a master's project which addresses this problem would really meet an existing need. During a series of meetings with Placement Center staff, specifications were developed for a microcomputer-based system of software and files to take over quite a few of the functions described. The system was designed to handle the interactions with the student in a simple, conversational style. It would keep a complete
file on all the students currently eligible to interview and a file of all upcoming employer interviews. It would print out a confirmation of an interview scheduled for the student, a schedule of interviews for the employer, and a letter to any student who had missed his or her interview time. The system is designed to be easily modified in order that it could grow with the needs of the Placement Center. It should be capable of being maintained by a student assistant, rather than requiring a professional programmer, and it should provide a reasonable level of data security against tampering or hardware failure.

The system described in this report attempts to meet all of these goals.
CONTENTS

LIST OF TABLES ........................................ viii
LIST OF FIGURES ........................................ ix

Chapter

I. INTRODUCTION ........................................ 1

II. SYSTEM REQUIREMENTS ................................. 7
    Development ........................................ 7
    Current Form ....................................... 9
    Future Requirements ............................... 14

III. SYSTEM DESCRIPTION ................................ 16
    Hardware Description ............................... 16
    Software Files ..................................... 18
    Programs ........................................... 26

IV. USE BY STUDENTS ..................................... 31

V. FUNCTIONS FOR EMPLOYER ............................. 41
    Use of EMPLUP Program ............................ 41
    Use of UPDAY Program ............................. 45

VI. FUNCTIONS FOR PLACEMENT CENTER ................ 48
    Use of STDNUP Program ............................ 48
    Use of DAYSCHED Program ........................ 51
    Use of DINGLTR Program ........................... 54
    Use of MSGCREA Program ........................... 56

VII. MODIFICATION AND TROUBLE SHOOTING .............. 58
    Use of FILETEST Program .......................... 58
    Use of CREAPG Program ............................ 64
    Use of AUTOEXEC.BAT .............................. 66
    Suggestions ........................................ 67
<table>
<thead>
<tr>
<th>VIII. CONCLUSIONS AND RECOMMENDATIONS</th>
<th>69</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX A STDNSRV--THE STUDENT SERVICE PROGRAM</td>
<td>72</td>
</tr>
<tr>
<td>Functional Description</td>
<td>72</td>
</tr>
<tr>
<td>Listing</td>
<td>80</td>
</tr>
<tr>
<td>APPENDIX B EMPLUP--EMPLOYER FILE UPDATING PROGRAM</td>
<td>92</td>
</tr>
<tr>
<td>Functional Description</td>
<td>92</td>
</tr>
<tr>
<td>Listing</td>
<td>98</td>
</tr>
<tr>
<td>APPENDIX C STDNUP--STUDENT FILE UPDATING PROGRAM</td>
<td>108</td>
</tr>
<tr>
<td>Functional Description</td>
<td>108</td>
</tr>
<tr>
<td>Listing</td>
<td>110</td>
</tr>
<tr>
<td>APPENDIX D UPDAY--INTERVIEW FILE CREATION PROGRAM</td>
<td>120</td>
</tr>
<tr>
<td>Functional Description</td>
<td>120</td>
</tr>
<tr>
<td>Listing</td>
<td>126</td>
</tr>
<tr>
<td>APPENDIX E DAYSCHED--INTERVIEW SCHEDULE WRITING PROGRAM</td>
<td>134</td>
</tr>
<tr>
<td>Functional Description</td>
<td>134</td>
</tr>
<tr>
<td>Listing</td>
<td>137</td>
</tr>
<tr>
<td>APPENDIX F PROGRAMS FOR STAFF USE</td>
<td>145</td>
</tr>
<tr>
<td>DINGLTR Functional Description</td>
<td>145</td>
</tr>
<tr>
<td>DINGLTR Listing</td>
<td>146</td>
</tr>
<tr>
<td>MSGCREA Functional Description</td>
<td>149</td>
</tr>
<tr>
<td>MSGCREA Listing</td>
<td>151</td>
</tr>
<tr>
<td>APPENDIX G MISCELLANEOUS PROGRAMS</td>
<td>155</td>
</tr>
<tr>
<td>FILETEST Functional Description</td>
<td>155</td>
</tr>
<tr>
<td>FILETEST Listing</td>
<td>156</td>
</tr>
<tr>
<td>CREAPG Functional Description</td>
<td>159</td>
</tr>
<tr>
<td>CREAPG 1 through 4 Listings</td>
<td>160</td>
</tr>
<tr>
<td>AUTOEXECUT BAT Listing</td>
<td>172</td>
</tr>
<tr>
<td>STARTUP Listing</td>
<td>172</td>
</tr>
<tr>
<td>EXECUTIV Listing</td>
<td>173</td>
</tr>
<tr>
<td>BIBILOGRAPHY</td>
<td>176</td>
</tr>
</tbody>
</table>
LIST OF TABLES

System Hardware ........................................... 17
Student File Contents. ................................. 19
Student Cross File Contents ......................... 20
Employer File Contents ............................... 22
Employer Cross File Contents ....................... 24
Employer Schedule File Contents ................. 25
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EXECUTIV--Menu</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>EXECUTIV--Choice of Basic</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>STDNSRV--Operating Message</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>STDNSRV--First Request</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>STDNSRV--SSN File Search</td>
<td>33</td>
</tr>
<tr>
<td>6</td>
<td>STDNSRV--No Such File</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>STDNSRV--Name Confirmation</td>
<td>34</td>
</tr>
<tr>
<td>8</td>
<td>STDNSRV--Telephone Number Check</td>
<td>34</td>
</tr>
<tr>
<td>9</td>
<td>STDNSRV--Number Correction</td>
<td>35</td>
</tr>
<tr>
<td>10</td>
<td>STDNSRV--Address Check</td>
<td>35</td>
</tr>
<tr>
<td>11</td>
<td>STDNSRV--Further Confirmations</td>
<td>36</td>
</tr>
<tr>
<td>12</td>
<td>STDNSRV--First Company</td>
<td>37</td>
</tr>
<tr>
<td>13</td>
<td>STDNSRV--Second Company</td>
<td>37</td>
</tr>
<tr>
<td>14</td>
<td>STDNSRV--A Company's Interviews</td>
<td>38</td>
</tr>
<tr>
<td>15</td>
<td>STDNSRV--Interview Confirmation</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>STDNSRV--Final Function</td>
<td>40</td>
</tr>
<tr>
<td>17</td>
<td>STDNSRV--Expected Companies</td>
<td>40</td>
</tr>
<tr>
<td>18</td>
<td>EMPLUP--Menu</td>
<td>42</td>
</tr>
<tr>
<td>19</td>
<td>EMPLUP--First Form</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>EMPLUP--Second Form</td>
<td>44</td>
</tr>
<tr>
<td>21</td>
<td>EMPLUP--File Message</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>Menu Item</td>
<td>Page</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>22</td>
<td>UPDAY--First Message</td>
<td>46</td>
</tr>
<tr>
<td>23</td>
<td>UPDAY--Employer Information</td>
<td>47</td>
</tr>
<tr>
<td>24</td>
<td>STDNUP--Menu</td>
<td>49</td>
</tr>
<tr>
<td>25</td>
<td>STDNUP--First Form</td>
<td>50</td>
</tr>
<tr>
<td>26</td>
<td>STDNUP--Second Form</td>
<td>51</td>
</tr>
<tr>
<td>27</td>
<td>STDNUP--Student File Number</td>
<td>52</td>
</tr>
<tr>
<td>28</td>
<td>DAYSCHED--Menu</td>
<td>52</td>
</tr>
<tr>
<td>29</td>
<td>DAYSCHED--Printout</td>
<td>53</td>
</tr>
<tr>
<td>30</td>
<td>DAYSCHED--Second Option</td>
<td>54</td>
</tr>
<tr>
<td>31</td>
<td>DINGLTR--Getting Started</td>
<td>55</td>
</tr>
<tr>
<td>32</td>
<td>DINGLTR--Sample Letter</td>
<td>56</td>
</tr>
<tr>
<td>33</td>
<td>MSGCREA--Starting Up</td>
<td>57</td>
</tr>
<tr>
<td>34</td>
<td>MSGCREA--Message Editing</td>
<td>57</td>
</tr>
<tr>
<td>35</td>
<td>FILETEST--Menu</td>
<td>60</td>
</tr>
<tr>
<td>36</td>
<td>FILETEST--Employer File</td>
<td>61</td>
</tr>
<tr>
<td>37</td>
<td>FILETEST--Employer Cross File</td>
<td>62</td>
</tr>
<tr>
<td>38</td>
<td>FILETEST--Student File</td>
<td>63</td>
</tr>
<tr>
<td>39</td>
<td>FILETEST--Student Cross File</td>
<td>64</td>
</tr>
<tr>
<td>40</td>
<td>FILETEST--Employer Schedule File</td>
<td>65</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

The growth of the University of Central Florida places an ever-increasing load upon the Placement Center, both in terms of the number of students on file and actively interviewing, and in the great number of prospective employers who schedule numerous visits for the purposes of job interviews. Currently, the Placement Center must use a number of forms and files to accomplish only this part of its task. The system is far from ideal, as the long lines of waiting students and the large expenditure of staff time attest.

Over a period of several months, in a number of meetings with Placement Center staff members, specifications for a microcomputer-based system of software and files was developed and is presented in Chapter 2 of this report.

This system is designed to handle most interaction with the students for interview scheduling, to keep a file of all students who are interviewing, to keep a file of all employers who are or intend to interview, and to provide
those employers with printed schedules of their interviews daily when they arrive at the University. A more detailed list of these functions appear below:

1. The system will interact with the student in a simple and conversational way in order to keep the student's file updated and allow him or her to schedule as many as five interviews with different companies on a given visit. It would give the student printed confirmation of these interviews.

2. The system will allow the staff to create, update, and delete the student files as required, utilizing two "forms" which would be filled out on screen in a manner similar to a printed form.

3. The system will allow the staff to create, update, and delete files of employers who would be visiting to interview students, using this same printed form approach.

4. The system will automatically open the scheduling of interviews 14 days in advance, creating files to store the required information, and deleting files for those employers who had interviewed in the past.

5. The system will allow the staff to produce a schedule of the day's interviews for the prospective employers.
6. The system will allow the staff to automatically send a letter to any student who misses a scheduled interview.

7. The system will assist the staff in creating messages which are automatically given the student when he or she signs up for an interview (e.g., "Be sure to bring a copy of your resume' when you come in for your interview.")

In order to accomplish these tasks, seven individual programs were written and are described in some detail later in this report. In addition to these programs, eight others of a subsidiary nature were written which are designed to:

1. Give a test listing of any software file used by the programs above for trouble shooting and program modification work.

2. Create the four "page" files which allow the staff to relate to the program as if it were printed forms.

3. Provide a series of automatically executed files which allow the software to start itself upon system power application, eliminating the need for the staff to be familiar with computer programming.

There are ten software files (five random and five sequential) which are kept by the system in order to perform its tasks. These serve the following purposes:
1. Keep a file of information on all students currently interviewing (accessed by the students' Social Security numbers.)

2. Keep a cross reference file between the Social Security numbers and the computer file numbers.

3. Keep a file of information on all employers who intend to interview in the next few months.

4. Keep a cross-reference file similar to the one used for students.

5. Keep a file of all open interviews—one file is needed for each day each company will be interviewing. These would be kept only for the period from 14 days prior to the first interview until after the last interview is completed.

6. Keep a file of the messages which have been recorded by the staff for the students' information.

7. Keep four files containing the information needed to draw the "pages" which staff persons use in updating the student and employer files.

The minimum hardware system required for this software is an IBM PC with double-sided drives, but an IBM PC-XT with a hard disk storage device is preferable. The files have been designed to be as compact as possible. For instance, the student information file requires only 256 "bytes"
(characters). The random files are all designed to fall on even disk sector boundaries for rapid execution (i.e., files are 16, 64, 256, and 512 bytes). With the present design maximum of 1,000 students, 500 employers, 200 active interview files and 20 message files, about 600 kilobytes of storage would be required for files alone.

It is intended that the programs eventually would all be compiled in order that the execution time be reduced. This would be done after two or three months of use of the system in parallel with the existing paperwork, in order to phase in the new system, to work out all the "bugs," and to make any modifications which might be required to assist the staff in easy use of the new system.

Modification and trouble shooting would be particularly easy for several reasons:

1. The modular nature of the software design allowing one function to be tinkered with without affecting other system tasks.

2. The use of the Basic language to ensure familiarity to the maximum number of programmers. It is anticipated that student assistants would be hired to do any programming chores the Placement Center might have.

3. The file testing program, which can be used as both a program design and a trouble shooting aid.
4. The extensive commenting within the program listings, explaining all major steps the programs perform, all variable names which are used, and the purposes of all files, etc.

Another important consideration for any computerized data base is security. In its final form, these programs would be very difficult to tamper with for several reasons. The student service routine would be compiled and the original language would not be available on the machine while the students are using it. The staff programs would be kept on a separate disk, and require a password for access. Student time on the machine would be limited automatically by the program, and by the presence of Placement Center staff. A backup copy of the files would be made by staff on a daily basis.

In summary, this system is designed to take over or to aid in several of the tasks currently performed by the Placement Center staff. In addition, it is intended to be easy to use, flexible, and easily updated. A minimum skill level is required of both the user and the programmer who maintains it.
CHAPTER 2

SYSTEM REQUIREMENTS

The requirements for this system were developed over a period of time during several meetings with the Placement Center staff. This chapter outlines the development of those requirements and the form for the system which was finally agreed upon. It also looks at some ways that the system might be refined and expanded in the future.

Development

The development of any system of computer hardware and software to accomplish tasks currently performed with paper forms and filing cabinets will be an evolutionary process. The office personnel must be a part of this development—they understand the current system, as well as its strong and weak points. It is possible, however, that office personnel may be somewhat reluctant to depart from methods with which they are familiar and from procedures which may no longer meet a need.

The thrust of the first meeting(s) must be for the system designer to become as familiar as possible with the current system. All of the currently used forms are
collected (preferably from files as filled out). The use of each of the items of information gathered must be understood. Some idea of the volume of data must be developed—for instance, how many students are in the files currently, and what data is now kept on each student.

One of the concerns expressed early on was whether the software for this system would become an "orphan." In other words, would there be support during the initial de-bugging and modifications, and could support be available over the long term, since the system would be expected to grow, to take on new tasks, and to be refined? This concern led to the adoption of the Basic programming language, the modularity of the software design, and use of extensive commenting, so that a relatively inexperienced student assistant familiar with Basic could do program work on the system.

In between meetings, interim system specifications were circulated to the staff in order to obtain reactions, new ideas, and to flag possible problems. It was decided that the system should use the approach of putting a "form" on the screen to be "filled out" by the staff in a way which would require little change in habit. The students would interact with a very user-friendly program, which would refer to itself as "I" and give a printed confirmation of the scheduled interview on the spot.
As the specifications for the software tasks became clear, the hardware which would be required was more obvious. A "stand alone" microcomputer system was desirable for maximum ease of use and flexibility. An inexpensive dot-matrix printer would be satisfactory for the job. At least two double-sided disk drives would be essential. The microcomputer would be free during part of the day for other office tasks such as word processing. It would be desirable to have as much commercial software as possible for these tasks. For future growth, a machine like the IBM PC seemed desirable, as it has become the standard for small business computing.

During later meetings important specifications like the size and number of student and employer files were tied down. The staff understood that these essential specifications were central to the design of the software, and could not be readily changed. These specifications were also chosen with an eye toward future growth.

Current Form

This section gives the requirements eventually decided upon for the Placement Center's system. It is extracted
from the final goals and specifications agreed to in a meeting with the staff of the Placement Center. More detailed specifications for hardware and software are in the next chapter.

It was concluded that:

1. Scheduling of interviews would be done directly by the student. The machine should interact with the student in a conversational, "user-friendly" style, and would be available a large number of hours (probably every weekday morning or afternoon) for student use.

2. In order that long lines would not develop on a single day when new interviewing opportunities are posted, the system should automatically open scheduling of interviews with an upcoming interviewer at some reasonable time (say ten days or two weeks before the interview visit). The student could schedule up to five interviews on a given day.

3. The student should be notified by the system of any upcoming interviewers who have expressed an interest in graduates in his or her field of study. This notice should be labeled "SUBJECT TO CHANGE--CHECK BACK." Some recruiters are initially scheduled without knowing what jobs they wish to fill. These shouldn't be listed.
4. The student would be given a printed form for each interview showing the employer, the date and time, and any other pertinent information. Up to two sentences of additional information for people interviewing with the particular employer would be printed. A disclaimer of some sort would be included to the effect that additions and changes are always possible. A notice that the student should notify the Career Planning and Placement Center in advance if unable to attend the scheduled interview should be printed on the form also.

   This form could be used to admit the student to the interview if required. Up to two sentences of additional information applicable to all persons interviewing may also be added at this point, for instance, "Be sure to bring a copy of your resume of the interviewer to keep."

5. The system would have the capability of giving a Placement Center staff person a list of students with telephone numbers, addresses, and other pertinent information in the event that a scheduled interview is cancelled or must be rescheduled.

6. When the interviewer arrives, the staff person could provide a printout of his or her schedule of
interviews which would include the student's name, major, grade point average, citizenship, address, graduation date, and other pertinent information.

7. The system would also have the capability of printing a schedule showing company name, times for interviews, day and month and whether the hour is open/scheduled.

8. In order for the integrity of the system database to be maintained, an access code would be required to perform any function other than student interview scheduling (the first four items in this list), and addresses or telephone number corrections which would be handled during interview scheduling.

9. Two categories of information (databases) must be maintained on hard disk or on two separate floppy discs: The first ("Employers") data; would contain the name of the company, employment location, position titles(s), information on the major, degree, requirement for citizenship and graduation dates the company is interested in, grade point average required and a schedule of interviewees with additional information they may be asked to provide. The second database ("Students") is a large file of students and graduates who are seeking work. This file would be
largely maintained by Placement Center staff with the exception of address and telephone numbers, etc. mentioned in item 8, above. This second file would be limited to 1000 persons unless "hard disk" storage medium is available.

10. In the event that the student disagrees with the information kept on him or her in the "Eligible Interviewee" file, a staff person will need to intervene. A form would be required for the student to request a change in the information because it is important to avoid giving the interviewer incorrect information (particularly about citizenship, major and grade point).

11. Upcoming interviews could be scheduled months in advance by a staff person and the dates, times, etc. changed as fresh information comes in without notification of change to users, until the date when scheduling is automatically opened. After scheduling has been opened, students whose schedules are affected can be notified according to a list generated (see item 5 above).

12. For students who don't meet the company's stated requirements, or who cannot make appointments because of scheduling conflicts, or because the company's
schedule is filled, a standby list could be maintained. The standby list would be kept by staff in order to notify people if more interviews are scheduled, or the list could be given to the interviewer for action.

13. A staff person could enter the information that a student did not show up for interview as scheduled. The system would automatically write a "no show letter."

14. When a student attempts to schedule an interview with a company, the program would automatically check his eligibility (i.e., grade point, graduation date, major, and citizenship) before scheduling the interview. A student who does not meet the criterion may choose to be placed on the standby list (see item 12 above).

Future Requirements

As has been previously stated, it is expected that the tasks performed by this system will need to change. New requirements will develop, new data will need to be filed, and problems will be discovered which need correction. For
these reasons the programs are modular--one can be worked upon without affecting the performance of another. In addition, more space is available in the data files for future needs. In fact, some of the data currently kept is not used.

It may be desirable in the future to have the system gather statistics on the students, the employers, or on some feature of the interview process. It could be that the program DINGLTR, which writes a letter to a student who has missed an interview, will be made completely automatic. Another future requirement might be that the system automatically notify all students affected when an interviewer cancels.

A program could be written to aid the process of "backing up" the primary storage medium. It would allow copying only files which may have been changed since the last copy was made.

In the area of hardware, a non-interruptable power supply would further protect the system database against errors. A clock-calendar card is a useful addition to any business microcomputer system. It might be desired to add a color monitor. A high quality impact printer for letter writing would definitely enhance the system's usefulness.
CHAPTER 3
SYSTEM DESCRIPTION

This chapter details the hardware and software file space specifications of the system. It also gives a general description of each of the programs used. A detailed section-by-section description of each program and program listing are in Appendices A through G.

Hardware Description

The reasons for the choice of the IBM PC were given in the last chapter. System components and their approximate (1985) prices are given in Table 2. The optional items include a clock/calendar card, which is handy on business systems, and three business software items which would be useful in any office.
<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (est.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Personal Computer XT with 256K RAM, 360K diskette drive, and 10 megabyte hard disk.</td>
<td>$3076.50</td>
</tr>
<tr>
<td>Monochrome/Parallel printer adapter card</td>
<td>175.00</td>
</tr>
<tr>
<td>Monochrome monitor</td>
<td>192.50</td>
</tr>
<tr>
<td>Printer---(Epson 80) (includes cable)</td>
<td>345.80</td>
</tr>
<tr>
<td>DOS 3.0 Program</td>
<td>45.50</td>
</tr>
</tbody>
</table>

**OPTIONAL ITEMS**

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost (est.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock/calendar card---(I/O plus 2)</td>
<td>325.00</td>
</tr>
<tr>
<td>AST Research</td>
<td></td>
</tr>
<tr>
<td>Color Monitor Adaptor Card</td>
<td>170.00</td>
</tr>
<tr>
<td>Color Monitor</td>
<td>476.00</td>
</tr>
<tr>
<td>Display Write III</td>
<td>244.00</td>
</tr>
<tr>
<td>Time Manager--an executive time scheduling program</td>
<td>70.00</td>
</tr>
</tbody>
</table>
Software Files

Tables 2 through 6 detail the contents of important software data files kept by the system. These are random access files stored on the floppy (or hard) disk mass storage device. Their sizes were chosen 25, 16, 64, 256, and 512 characters or bytes so that the files would fall on even disk sector boundaries for faster access times.

The student file, which is called "STUDENTS" in the software, contains all the appropriate information on the individual students who are actively interviewing. The second telephone number descriptor may be as follows: 1) work phone, 2) at parents' home, 3) at a friend's home. The state should be abbreviated: Florida would FL, etc. The graduation term would be: 1) Fall, 2) Spring, or 3) Summer. The graduation year is the last two digits of the year of expected graduation. The major code is a four digit code used by the University to indicate the student's major. The citizenship descriptor indicates: 1) U.S. citizen, 2) permanent visa, or 3) a student visa. The degree descriptor should be: 1) if bachelor's, 2) if master's, 3) if doctoral, and 4) if a special certificate. The grade point average is given as of the term and year computed using the formats given above. The interview history section is designed for future use.
TABLE 2
STUDENT FILE CONTENTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>15</td>
</tr>
<tr>
<td>First Name</td>
<td>14</td>
</tr>
<tr>
<td>Middle Initial</td>
<td>1</td>
</tr>
<tr>
<td>Social Security Number</td>
<td>9</td>
</tr>
<tr>
<td>Telephone Number (1)</td>
<td>10</td>
</tr>
<tr>
<td>Telephone Number (2)</td>
<td>10</td>
</tr>
<tr>
<td>Descriptor for Second Telephone</td>
<td>1</td>
</tr>
<tr>
<td>Street Address (Local)</td>
<td>20</td>
</tr>
<tr>
<td>City</td>
<td>15</td>
</tr>
<tr>
<td>State</td>
<td>2</td>
</tr>
<tr>
<td>Zip Code</td>
<td>5</td>
</tr>
<tr>
<td>Street Address (Permanent)</td>
<td>20</td>
</tr>
<tr>
<td>City</td>
<td>15</td>
</tr>
<tr>
<td>State</td>
<td>2</td>
</tr>
<tr>
<td>Zip Code</td>
<td>5</td>
</tr>
<tr>
<td>Country</td>
<td>10</td>
</tr>
<tr>
<td>Graduation Term</td>
<td>1</td>
</tr>
<tr>
<td>Graduation Year</td>
<td>2</td>
</tr>
<tr>
<td>Major Code</td>
<td>4</td>
</tr>
<tr>
<td>Citizenship Descriptor</td>
<td>1</td>
</tr>
<tr>
<td>Degree Descriptor</td>
<td>1</td>
</tr>
<tr>
<td>Grade Point Average</td>
<td>3</td>
</tr>
<tr>
<td>Term Computed</td>
<td>1</td>
</tr>
<tr>
<td>Year Computed</td>
<td>2</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE ONE</strong></td>
<td>154</td>
</tr>
<tr>
<td>Interview History Date</td>
<td>6</td>
</tr>
<tr>
<td>Interview History Code</td>
<td>4</td>
</tr>
<tr>
<td><em>(Interview History Repeats 5 More Times)</em></td>
<td>50</td>
</tr>
<tr>
<td>Unused File Space</td>
<td>27</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE TWO</strong></td>
<td>102</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>256</td>
</tr>
</tbody>
</table>
The student cross file, which is called "STDNCROS" in the software, contains information which allows quicker access to the student files. Normally when the software is hunting for a student by social security number, it will find the student in the cross file first, then, since the file number in the student file is the same as in the cross file, the student file can be read quickly. The cross file contains another important item—the date of last use. Whenever the student file is used, this date will be made current. If it becomes necessary in the future to purge inactive students, it can be easily done by referring to this date in all files.

TABLE 3
STUDENT CROSS FILE CONTENTS

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Security Number</td>
<td>9</td>
</tr>
<tr>
<td>Date of Last Use</td>
<td>6</td>
</tr>
<tr>
<td>Unused File Space</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16</strong></td>
</tr>
</tbody>
</table>
The employer file, which is called, "EMPLYRS" in the software, contains all the information on the employer and the interview schedule except the identification of students who have made appointments. Most of the data (such as the state, zip code, etc.) are filed similarly to the student file with these additions. Messages to students can contain a two-digit number referring to information in the message file which the employer desires the students to have. Messages such as "Be sure to bring a copy of your resume," may be included in this file. The employer may select which term and year of graduation and what degree and major he or she desires to interview. Space is provided to store up to five interview dates, with the number of interviewers (up to six), and the number of morning and afternoon interviews (up to nine each) required. The length of the interview can be: 1) 30 minutes, 2) 45 minutes, or 3) one hour. The software will automatically compute the interview schedules based on this information and on the morning and afternoon start times. Afternoon start time must be in military time, i.e., 1330 would be 1:30 PM. Twenty characters are allocated for future needs.
### TABLE 4

**EMPLOYER FILE CONTENTS**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company Name</td>
<td>25</td>
</tr>
<tr>
<td>Division</td>
<td>25</td>
</tr>
<tr>
<td>Street Address</td>
<td>25</td>
</tr>
<tr>
<td>City</td>
<td>15</td>
</tr>
<tr>
<td>State</td>
<td>2</td>
</tr>
<tr>
<td>Zip Code</td>
<td>9</td>
</tr>
<tr>
<td>Country</td>
<td>10</td>
</tr>
<tr>
<td>Contact Person's Last Name</td>
<td>15</td>
</tr>
<tr>
<td>First Name</td>
<td>14</td>
</tr>
<tr>
<td>Middle Initial</td>
<td>1</td>
</tr>
<tr>
<td>Title</td>
<td>25</td>
</tr>
<tr>
<td>Telephone Number</td>
<td>8</td>
</tr>
<tr>
<td>Extension</td>
<td>9</td>
</tr>
<tr>
<td>Messages to Students</td>
<td>2</td>
</tr>
<tr>
<td>(Messages Repeat 4 More Times)</td>
<td>8</td>
</tr>
<tr>
<td>Terms to be Interviewed</td>
<td>2</td>
</tr>
<tr>
<td>Years to be Interviewed</td>
<td>2</td>
</tr>
<tr>
<td>(Interview Repeats 3 More Times)</td>
<td>9</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE ONE</strong></td>
<td><strong>205</strong></td>
</tr>
<tr>
<td>Major Codes To Be Interviewed</td>
<td>4</td>
</tr>
<tr>
<td>(Major Repeat 9 More Times)</td>
<td>36</td>
</tr>
<tr>
<td>Degree Sought</td>
<td>1</td>
</tr>
<tr>
<td>(Degree Repeats 3 More Times)</td>
<td>3</td>
</tr>
<tr>
<td>Citizenship Required</td>
<td>1</td>
</tr>
<tr>
<td>Location of Work</td>
<td>25</td>
</tr>
<tr>
<td>Minimum Grade Point Average</td>
<td>3</td>
</tr>
<tr>
<td>Position Title(s)</td>
<td>50</td>
</tr>
<tr>
<td>Number of Interviewers</td>
<td>1</td>
</tr>
<tr>
<td>Date of Interview</td>
<td>6</td>
</tr>
<tr>
<td>Morning Interviews</td>
<td>1</td>
</tr>
<tr>
<td>Afternoon Interviews</td>
<td>1</td>
</tr>
<tr>
<td>(Interviews Repeat 4 More Times)</td>
<td>36</td>
</tr>
<tr>
<td>Length of Interview</td>
<td>1</td>
</tr>
<tr>
<td>Morning Start Time</td>
<td>4</td>
</tr>
<tr>
<td>Afternoon Start Time</td>
<td>4</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE TWO</strong></td>
<td><strong>117</strong></td>
</tr>
</tbody>
</table>
Table 4 (continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Number of First Schedule</td>
<td>4</td>
</tr>
<tr>
<td>Number of Open Interviews by Hour</td>
<td>18</td>
</tr>
<tr>
<td>(Schedule Repeats 4 More Times)</td>
<td></td>
</tr>
<tr>
<td>Unused File Space</td>
<td>88</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
</tr>
<tr>
<td>SUBTOTAL PAGE THREE</td>
<td>130</td>
</tr>
<tr>
<td>TOTAL</td>
<td>512</td>
</tr>
</tbody>
</table>

The employer cross file, which is called "EMPLCROS" in the software, operates in a similar way to the student cross file. Here the company name and division are the items which will be searched for by programs. As with the student cross file, the file number of the employer file is the same as the file number for its cross file. Three other important items that are used in the cross file: the first and last interview dates are used in software as a first check to see if the particular employer may be interviewing on a desired date, if the files are old and may be subject to deletion, or if the interviews for the particular employer are coming up in the near future, and interview files need to be opened. The third item is the interviews open indicator. If a "1" is stored in this location, it indicates that interview files have already been opened and students may already be scheduled for the dates and times shown in this particular employer file.
TABLE 5
EMPLOYER CROSS FILE CONTENTS

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company Name</td>
<td>25</td>
</tr>
<tr>
<td>Division</td>
<td>25</td>
</tr>
<tr>
<td>First Interview Date</td>
<td>6</td>
</tr>
<tr>
<td>Last Interview Date</td>
<td>6</td>
</tr>
<tr>
<td>Interviews Open Indicator</td>
<td>1</td>
</tr>
<tr>
<td>Unused File Space</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>64</td>
</tr>
</tbody>
</table>

The employer schedule file, which is called "EMPLSCHE" in the software, contains all the information for a single day of an employer's interviewing. Each of the nine morning and afternoon interview times may have up to six interviewers, so that a total of 18 x 6 or 108 interviews could be scheduled in one schedule file. To obtain a student file number from this file, the software takes four times the interviewer number plus 24 times the interview hour and adds 216 if it is an afternoon interview, it then subtracts 27. This gives the location of the 4 digit student file number for the student scheduled. The remaining information in the file is a copy (from the employer file) of date and time information for the particular interview day, and the employer file number to which this day's schedule refers. There are 58 unused spaces in the file.
TABLE 6
EMPLOYER SCHEDULE FILE CONTENTS

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student Files for 6 Interviewers</td>
<td>24</td>
</tr>
<tr>
<td>(Files Repeat for 8 more Morning Interview Schedules)</td>
<td>192</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE ONE</strong></td>
<td>216</td>
</tr>
<tr>
<td>(Files Repeat for 9 Afternoon Interview Possibilities)</td>
<td>216</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE TWO</strong></td>
<td>216</td>
</tr>
<tr>
<td>Number of Interviewers</td>
<td>1</td>
</tr>
<tr>
<td>Date of Interview</td>
<td>6</td>
</tr>
<tr>
<td>Morning Interviews</td>
<td>1</td>
</tr>
<tr>
<td>Afternoon Interviews</td>
<td>1</td>
</tr>
<tr>
<td>Length of Interview</td>
<td>1</td>
</tr>
<tr>
<td>Morning Start Time</td>
<td>4</td>
</tr>
<tr>
<td>Afternoon Start Time</td>
<td>4</td>
</tr>
<tr>
<td>Employer File Number</td>
<td>4</td>
</tr>
<tr>
<td>Unused File Space</td>
<td>58</td>
</tr>
<tr>
<td><strong>SUBTOTAL PAGE THREE</strong></td>
<td>80</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>512</td>
</tr>
</tbody>
</table>

Two more types of files need to be mentioned. The employer and staff message file, called "MESSAGES," and the four files used in drawing "forms" on the screen, called "PAGEONE," "PAGETWO," "PAGETHRE," and "PAGEFOUR." Messages have been mentioned before. They can be recorded by the staff and displayed automatically to the student when an interview is scheduled. Up to 99 messages of 256 characters each could be filed, depending upon availability of storage space. Message numbers are recorded in the employer file for display when an interview is scheduled.
The form drawing files, "PAGEONE," etc., store the screen locations and characters to print at those locations in order that a form may be completed. They also store the locations to be read from the screen after the data has been filled in. The first two pages are used by the student file updating program, and the last two are used in the employer file program. These files are discussed further in the next section and in Appendix G.

Programs

This section introduces the names and purposes of the various programs. In the next three chapters a detailed description of the use of the programs is presented, and a detailed functional description of the operation of each program is presented in the appendix before the program listing.

All programs may be accessed through the master program, "EXECUTIV," which is automatically called up when power is applied to the machine. After the password is entered, the following menu appears (see Figure 1).
WELCOME TO THE UNIVERSITY OF CENTRAL FLORIDA PLACEMENT CENTER INTERVIEW SCHEDULING AND STUDENT RECORD KEEPING SYSTEM. THESE FUNCTIONS ARE AVAILABLE...

1) START THE DAY BY UPDATING INTERVIEW FILES
2) EDIT OR CREATE NEW STUDENT FILES
3) EDIT OR CREATE NEW EMPLOYER SCHEDULES
4) SEND A LETTER TO STUDENTS WHO MISSED AN INTERVIEW
5) PRINT OUT A SCHEDULE OF EMPLOYERS INTERVIEWING
6) EDIT OR CREATE MESSAGES FOR THE STUDENTS
7) ENTER 'BASIC' FOR EDITING PROGRAMS
8) RUN THE STUDENT SERVICE ROUTINE
9) RUN THE FILE TESTING PROGRAM

* INDICATES YOU MUST RUN 'EXECUTIV' (DRIVE A) TO RETURN TO SYSTEM

CHOOSE (1 TO 9) PLEASE.

Figure 1: EXECUTIV--MENU

If the user wishes to know the names of the various programs and files available, he or she would strike the number seven key and the following would appear (Figure 2).

YOU ARE NOW IN 'BASICA/F:3/S:512' WHICH MEANS YOU MAY HAVE UP TO THREE FILES OF UP TO 512 BYTES OPEN AT ONE TIME

DISK A INCLUDES: AUTOEXEC.BAT, STARTUP.BAS, EXECUTIV.BAS
CREAPG1.BAS, AND CREAPG2.BAS

DISK B INCLUDES: STDNUP.BAS, MSGCREA.BAS, FILETEST.BAS,
CREAPG4.BAS, STDNSRV.BAS, DAYSCHED.BAS, EMPLUP.BAS,
CREAPG1.BAS, UPDAY.BAS, AND DINGLTR.BAS

DISK A ALSO INCLUDES: SEQUENTIAL FILES PAGEONE AND PAGETWO
DISK B ALSO INCLUDES: SEQUENTIAL FILES PAGETHRE AND PAGEFOUR
AS WELL AS RANDOM FILES STUDENTS, STDNCROS, EMLYRS, EMPLCROS,
AND EMPLSCHE

Figure 2: EXECUTIV-Choice of Basic
The program "STDNSRV" is the program which will be on the machine when the student comes in to schedule an interview. It will schedule up to five interviews, giving a printed confirmation and any messages which may be required. It will also allow the student to update the local address and telephone number. It will inform the student of five important items of information in his or her file which cannot be directly updated: the graduation date, major code, citizenship, degree, and grade point average. If the student wishes to change these items, a form must be returned, as staff confirmation is necessary. Finally, the program will offer the student a list of interviewers who may be interested in talking to him or her within the next 30 days.

"STDNUP" is used by the staff to create the original student file for any student (or graduate) authorized to interview prospective employers. It is also used to update the student files and to delete them as required. All the items in the "STUDENT" file are accessible through this program and appear on the two forms recorded in "PAGEONE" and "PAGETWO."

"EMPLUP" serves the same purpose for the employer files that "STDNUP" does for the student files. They are very similar in operation. After interview schedule files
have been opened it is hazardous to change anything to do with the times, dates, etc., because students may have already scheduled an interview. A warning message to this effect is given.

At the start of every interview day, the program "UPDAY" is called by a staff person. All employers will be checked to see if interviews need to be opened for them. If the first interview day listed is less than 14 days in the future, the staff person will be asked whether to set up the interview files for that company. After the files are opened, the students will be able to sign up for interviews. A second function is to see if the last interview day is in the past. If it is, the program will ask the staff person whether to erase the interview files for that company and if so, whether to erase the employer file too, thus saving file space. This program deals with the employer file, the employer cross file and the interview schedule files.

"DAYSCHED" is the program used by the staff to produce printed interview schedules. It has two modes of operation; in the first it prints out the schedule for all employers for a single day, and in the second mode it prints all of a given employer's schedules for up to five interview days. In both cases it prints the student's
name, social security number, grade point average, graduation date, citizenship status, the time of the interview, and interviewer number. This program accesses all five main files.

"DINGLTR" writes a letter to a student who has missed an interview. It needs to be given the interview date, the company name and the student's file number. The complete letter is then automatically printed out.

The program "MSGCREA" does the creation of the messages which may be given to the students upon scheduling an interview. It can also revise and delete existing messages.

Several more programs are listed in Appendix G and will only be mentioned here. "FILETEST" is used to write a copy of any of the five important files to the screen for trouble shooting and system modification work. "CREAPG1," "CREAPG2," CREAPG3," and "CREAPG4" are the programs which create the files ("PAGEONE," etc.) used in form drawing. If any change is required to the forms, these programs are easily revised, and run to create the new page file. The "AUTOEXECUTE.BAT" file and "STARTUP" are used to get the password and call up "EXECUTIV" when power is applied to the system.
CHAPTER 4

USE BY STUDENT

This program is an interactive and user friendly one which should prove easily used, even to the student who has not used a computer before. When the student arrives at the terminal, the words "strike any key" appear on the screen. Upon pressing any key, the computer displays the message shown in Figure 3. Notice that the computer calls itself "I" and tells the student what is needed or what it is doing at every step of the way.

WELCOME TO THE UNIVERSITY OF CENTRAL FLORIDA'S PLACEMENT CENTER

I'M HERE TO AID YOU IN SIGNING UP FOR PLACEMENT INTERVIEWS. I'LL ALSO HELP YOU TO UPDATE YOUR FILE -- ADDRESS, TELEPHONE NUMBER, ETC.

ARE YOU READY TO CONTINUE? (Y OR N)

Figure 3: STDNSRV--Opening Message

After the student indicates readiness to continue by pressing "y", the display shown in Figure 4 is generated.
Because of the possibility that the student will mistype their number, the computer will check the number as shown in Figure 5. If unsuccessful in finding a student file with the given social security number, the computer responds with the display of Figure 6, and if successful it uses the display in Figure 7 to confirm the name of the student.

**GOOD! FIRST I'LL NEED YOUR SOCIAL SECURITY NUMBER SO I CAN LOOK UP YOUR FILE. AFTER THAT, I'LL GIVE YOU AN OPPORTUNITY TO MAKE CORRECTIONS IF YOU WOULD LIKE TO.**

PLEASE ENTER YOUR SOCIAL SECURITY NUMBER EXACTLY LIKE THE ONE I'VE SHOWN HERE: 123-45-6789 --THEN PRESS RETURN (← )
YOUR NUMBER ——>

**Figure 4: STDNSRV--First Request**

**GOOD! FIRST I'LL NEED YOUR SOCIAL SECURITY NUMBER SO I CAN LOOK UP YOUR FILE. AFTER THAT, I'LL GIVE YOU AN OPPORTUNITY TO MAKE CORRECTIONS IF YOU WOULD LIKE TO.**

PLEASE ENTER YOUR SOCIAL SECURITY NUMBER EXACTLY LIKE THE ONE I'VE SHOWN HERE: 123-45-6789 --THEN PRESS RETURN (← )
YOUR NUMBER ——> 100-00-0000

JUST TO BE SURE I'VE GOT IT RIGHT, DID YOU GIVE ME SOCIAL SECURITY NUMBER 100-00-0000? (Y OR N)

ALRIGHT...NOW I NEED TO LOOK FOR YOUR FILE.
JUST A MOMENT, PLEASE.

**Figure 5: STDNSRV--SSN File Search**
GOOD! NOW WE CAN GET DOWN TO BUSINESS. FIRST, Eduardo
LETS CHECK SOME OF THE INFORMATION IN YOUR FILE.
FOR ANY ITEM THAT'S WRONG TYPE 'N', TYPE THE RIGHT
INFORMATION BELOW IT. THEN PRESS RETURN ( — )

YOUR HOME (LOCAL) TELEPHONE NUMBER—
111-222-3333 ...IS THIS NUMBER CORRECT? (Y OR N)

Figure 8: STDNSRV--Telephone Number Check

GOOD! NOW WE CAN GET DOWN TO BUSINESS. FIRST, Eduardo
LETS CHECK SOME OF THE INFORMATION IN YOUR FILE.
FOR ANY ITEM THAT'S WRONG TYPE 'N', TYPE THE RIGHT
INFORMATION BELOW IT. THEN PRESS RETURN ( — )

YOUR HOME (LOCAL) TELEPHONE NUMBER—
111-222-3333 ...IS THIS NUMBER CORRECT? (Y OR N)
...—...—... (NEW NUMBER)

Figure 9: STDNSRV--Number Correction

GOOD! NOW WE CAN GET DOWN TO BUSINESS. FIRST, Eduardo
LETS CHECK SOME OF THE INFORMATION IN YOUR FILE.
FOR ANY ITEM THAT'S WRONG TYPE 'N', TYPE THE RIGHT
INFORMATION BELOW IT. THEN PRESS RETURN ( — )

YOUR LOCAL ADDRESS (STREET/CITY/STATE/ZIP)
007 Bond Street
Jamestown
FA
33333

...DO I HAVE YOUR ADDRESS RIGHT? (Y OR N)

Figure 10: STDNSRV--Address Check
Five more important items are in the student's file and will be confirmed. The student may change these only by completing a form so that the staff may verify the information given. See Figure 11.

_SOME OTHER INFORMATION WHICH I HAVE ON FILE IS SHOWN BELOW. IN ORDER TO MAKE ANY CHANGES IN IT YOU WILL NEED TO FILL OUT A FORM WHICH YOU WILL FIND NEAR THIS CONSOLE. (A STAFF PERSON MUST VERIFY THE INFORMATION AND PUT IT ON FILE)._

1) EXPECTED (OR ACTUAL) GRADUATION IN SPRING OF 1985
2) YOUR MAJOR CODE IS LISTED AS 0930. YOU MAY CHECK IT AGAINST A LIST OF MAJORS IN THE SCHOOL CATALOG OR AGAINST A LIST WHICH MAY BE FOUND NEAR THIS CONSOLE.
3) I SHOW YOU AS HAVING ONLY A WORK PERMIT.
4) MY FILES INDICATE YOU HAVE OR ARE SEEKING A BACHELORS DEGREE.
5) AND IN THE FALL TERM OF 1984, YOUR GRADE POINT AVERAGE WAS 3

ARE YOU READY TO CONTINUE? (Y OR N)

Figure 11: STDNSRV--Further Confirmations

If the student were to leave the computer in the middle of a session, a timing function will, after a minute or so, reset the program and begin displaying the "strike any key" message.

After confirming the information above, the program asks the student whether they wish to schedule an interview.
and proceeds into the scheduling routine with the display of the first company for whom interviews are open. All interviewing companies will be displayed, even if the student does not apparently match their needs on one of the five items in Figure 11. This allows the student to ask to have his or her name put on a waiting list, etc. The first company, (Figure 12) is not interviewing the student's major, but the second company in Figure 13 would be interested in interviewing the student.

**Company Name:** Bogus Oil

**Division:** Esotericprofiticus

---Is not interviewing your major.

I am sorry. If you are convinced that you are qualified, you will need to speak to a staff person about it. Perhaps you can sign up on a waiting list.

Are you ready to check the next company? (Y/N)

![Figure 12: STDNSRV--First Company](image)

**Company Name:** Test Company

**Division:** Major Profits

---Would be interested in talking to you. Do you wish to set up an interview with them? (Y/N)

![Figure 13: STDNSRV--Second Company](image)
If the student does choose to seek an interview with the second company, the program will print a complete schedule of interviews available with that company (Figure 13). Note that 1330 in the afternoon of the third day and 0800 on the morning of the fourth day do not appear in the print because these two time slots have already been filled for all the company's interviewers scheduled on those times.

<table>
<thead>
<tr>
<th>DAY CODE</th>
<th>DATE</th>
<th>TIMES AVAILABLE ON THAT DATE</th>
<th>AM THEN PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>02-20-85</td>
<td>800 830 900 930 1000 1030</td>
<td>1200 1230 1300 1330 1400 1430</td>
</tr>
<tr>
<td>2</td>
<td>03-01-85</td>
<td>800 830 900 930 1000 1030</td>
<td>1200 1230 1300 1330 1400 1430</td>
</tr>
<tr>
<td>3</td>
<td>03-15-85</td>
<td>800 830 900 930 1000 1030</td>
<td>1200 1230 1300 1330 1400 1430</td>
</tr>
<tr>
<td>4</td>
<td>03-30-85</td>
<td>800 830 900 930 1000 1030</td>
<td>1200 1230 1300 1330 1400 1430</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WHICH DAY DO YOU WANT? (ENTER 'DAY CODE' 1 TO 5)...

Figure 14: STDNSRV--A Company's Interviews

After putting the interview dates and times on display, the program determines which day and hour the student wishes to schedule, files the interviewer schedule, and gives the student a confirmation of it as shown in Figure 15.
INTERVIEW SCHEDULED FOR Eduardo L. Contraro
SOCIAL SECURITY NUMBER 100-20-3000

COMPANY NAME: Test Company
DIVISION: Major Profits

DATE: 03-30-85

IN THE MORNING AT 9:30. PLEASE BE ON TIME.

THE INTERVIEWS HAVE ASKED THAT YOU BE GIVEN THE FOLLOWING INFORMATION:

MESSAGE #1 IS:
The student is advised that employers usually request a copy of their resume during the interview.

MESSAGE #2 IS:
Please be advised that this is a long boring message to tell you that you must do or not do something that you wished you could not do or do but knew all along that you would or would not do it anyway. Right?

DO YOU WISH TO SCHEDULE ANOTHER INTERVIEW? (Y/N)

Figure 15: STDNSRV--Interview Confirmation

As finally implemented, the confirmation would come from a printer beside the computer. Two messages are also printed below the confirmation. These may have been left by either the employer or the Placement Center staff as a special instruction to the student.

Upon showing the student all of the companies currently open for interviews, the program displays the message in Figure 16, which indicates the other important function the program performs--giving a listing of companies which are expected to interview within the next 30 days (see Figure 17). With the listing the program is
done, the student is warned that the information may change, and told that their session is complete.

This completes the use of STDNSRV. The files which it uses are STUDENTS, STDNCROS, EMLYRS, EMPLCROS, MESSAGES, and EMPLSCH. They were detailed in the previous chapter. For a complete listing of STDNSRV, see Appendix A.

THERE ARE NO MORE COMPANIES INTERVIEWING AT THIS TIME.

DO YOU WISH A LISTING OF COMPANIES WHO ARE INTERVIEWING DURING THE NEXT MONTH?

Figure 16: STDNSRV--Final Function

DIVISION: Esotericprofiticus
IS EXPECTED ON THE FOLLOWING DATES...
1 ) 03-01-85
2 ) 03-15-85
3 ) 03-30-85
4 ) 04-01-85
5 ) 04-15-85

COMPANY NAME: Test Company
DIVISION: Major Profits
IS EXPECTED ON THE FOLLOWING DATES...
1 ) 02-20-85
2 ) 03-01-85
3 ) 03-15-85
4 ) 03-30-85
5 ) --

THAT’S ALL THE INTERVIEWERS THAT I KNOW OF. STAY IN TOUCH FOR LATER INFORMATION, BECAUSE IT DOES CHANGE QUITE OFTEN.

YOUR SESSION IS NOW COMPLETE. IT HAS BEEN A PLEASURE TO SERVE YOU. PLEASE LET THE NEXT PERSON TAKE A BEAT. THANK YOU.

Figure 17: STDNSRV--Expected Companies
CHAPTER 5
FUNCTIONS FOR EMPLOYER

The use of two programs, EMPLUP and UPDAY, is described in this chapter. These programs are used by the staff to enter all data about an employer who intends to interview, to keep this data up to date as the interview dates draw near, to open interview schedules for the employer at the appropriate time, and to delete those interview schedules and employer files after the interviewing is complete. Up to six interviewers on five different interview days may be scheduled for 30-, 45-, or 60-minute interview time slots. If more dates or interviewers are to be scheduled, a separate employer file for the extra information would be required. The tentative dates and times may be filed several months in advance and changed as new data becomes available up to the time that interview schedules are actually opened (usually at about 14 days prior to the first interview date).

Use of EMPLUP Program

All programs used by the staff can be reached through the EXECUTIV program, which is automatically called when
system power is applied as discussed at the end of Chapter 3. The first message displayed by EMPLUP is the menu shown in Figure 18.

**THIS PROGRAM EDITS OR CREATES NEW EMPLOYER FILES**

**DO YOU WISH TO:**
1) EDIT/DELETE AN EXISTING EMPLOYER FILE?
2) CREATE A NEW EMPLOYER FILE?
3) RETURN TO THE MAIN MENU?
**CHOOSE (1,2,OR 3) PLEASE =**

![Figure 18: EMPLUP--Menu](image)

All data entry in EMPLUP is done by having the staff "fill out a form" which is displayed on the screen as if it were a printed form. Each blank is accessed in turn by entering a carriage return after the last blank has been completed. The back space and editing keys (up arrow, back arrow, etc.) are also active, and can be used to move back to an area of the form if a mistake is discovered. If the staff person desires to discard the page and return to the previous menu, this can be done by entering menu$ and return. See Figure 19. No commas may be used within the data fields.
ENTER INFORMATION WITHIN APPROPRIATE FIELDS USING <RETURN> TO MOVE BETWEEN FIELDS. DO NOT USE COMMAS! YOU ALSO MAY ENTER MENU* AND <RETURN> TO DISCARD THE INFORMATION ENTERED AND RETURN TO THE LAST MENU.

<table>
<thead>
<tr>
<th>COMPANY NAME</th>
<th>CONTACT: (LAST)</th>
<th>(FIRST)</th>
<th>(M/I)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIVISION</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>PHONE</th>
<th>EXTENSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(STREET)</td>
<td>(STINT)</td>
<td>(EXT)</td>
</tr>
<tr>
<td>(CITY)</td>
<td>(STATE)</td>
<td></td>
</tr>
<tr>
<td>(ZIP CODE)</td>
<td>(COUNTRY)</td>
<td></td>
</tr>
</tbody>
</table>

MESSAGES TO INTERVIEWEES:

BRADUATION DATES (TERM/YEAR):

(1=FALL, 2=SPRG, 3=SUMR, 4=ALL)

Figure 19: EMPLUP--First Form

Upon completion of the first four, the program asks the operator whether they wish to file the data as it is shown, to discard the data and return to the previous menu, or to further edit the data. If the file were one which had been previously created, the data originally filed would be shown on the form with the same choices available. If the data displayed is correct, the operator chooses to file it as shown and the computer displays the second form as shown in Figure 20.
ENTER INFORMATION WITHIN APPROPRIATE FIELDS USING <RETURN> TO MOVE BETWEEN FIELDS. DO NOT USE COMMAS! YOU ALSO MAY ENTER MENU* AND <RETURN> TO DISCARD THE INFORMATION ENTERED AND RETURN TO THE LAST MENU.

<table>
<thead>
<tr>
<th>MAJOR CODES TO BE INTERVIEWED</th>
<th>INTVWRS</th>
<th>DATE</th>
<th>NO. OF INTVWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ALL MAJ. = 000, ALL ENGR. = 0900, ETC)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DEGREES CIGNIZENSHIP

1) BACH. 3) DOCTOR 1) REG. U.S.A. 
2) MASTER 4) CERTF. 2) PERM VISA 
3) WORK PERMIT 3) WORK PERMIT 

WORK LOCATION(S) MINIMUM GPA:

POSITION NAME(S):

LENGTH OF INTERVIEW

(1=30, 2=45, 3=60 MIN) |

START TIMES AM PM 

(HR/MIN) |

Figure 20: EMPLUP--Second Form

Some of the data fields require further explanation. On the first form, "messages to interviewees" indicates the two-digit numbers of the messages stored by the program MSGCREA for the student to read when they sign up for an interview. If a new message is desired by an employer, it would first be entered in MSGCREA so that its number would be known. The graduation dates are entered as one-digit numbers (1 indicates fall, etc.). Note that the number 4 is entered to indicate that any graduate of the year will be interviewed. Of course, year is the last two digits of the year specified.
On the second form the four-digit major codes for all acceptable majors are entered. Note that while an 0833 might be used to indicate a major in Mathematics Education, an 0800 would indicate all graduates from the College of Education would be considered, and 0000 indicates that all majors from all colleges would be interviewed. The level of the degree, the citizenship requirement, and minimum grade point average are equally important entries, as they are used to screen out unwanted interviewees.

Under interviewers, a one-digit number indicates the number of interviewers (1 to 6) on the given date. "Number of interviews" indicates the number which should be scheduled in the morning and afternoon, and must be figured in conjunction with the length of the interview and the morning and afternoon start times. Note that the start time should be expressed in military time (i.e., 09:30, 14:45, etc.).

When the operator finishes with the second form, the same options are given as when the first form was completed. If the data is good, or after the data has been updated, it is filed, and the computer responds with Figure 21, reminding the staff person to keep the file number for future reference. The program then returns to the main menu shown in Figure 18.
THE EMPLOYER WILL BE GIVEN FILE NUMBER 4 -- YOU WILL WANT TO RETAIN THE NUMBER IN YOUR RECORDS.

>>> PRESS ANY KEY TO CONTINUE <<<

Figure 21: EMPLUP--File Message

This completes discussion of the use of EMPLUP. The files which it uses are EMPLYRS, EMPLCROS, PAGETHRE, and PAGEFOUR. They were detailed in Chapter 3. For a complete listing of EMPLUP, see Appendix B.

Use of UPDAY Program

UPDAY has three functions: If interviews have not been opened and the first interview date is 14 days or less in the future, it can be used to open interviews and create the interview files: If the last interview date has passed, it can be used to delete the interview files. Its third function if desired is to delete the employer file for which the interviews were scheduled. When called, UPDAY responds with the message shown in Figure 22.
This program is 'UPDAY'. It will help you open interviews for upcoming employers. You will need to decide whether to open each employer's interview schedule as it is called out. You will also be asked whether to delete old interview and employer files.

Do you wish to continue? (Y/N)

Figure 22: UPDAY--First Message

If the staff member chooses to continue, the program immediately begins sorting through the employer cross file looking for employers which may need their interview files opened or deleted. Since it is required that the day's date be entered when the system power is turned on, no further data is necessary to the program. When UPDAY finds an employer which may need attention it gives the information in the form shown in Figure 23.
EMPLOYER FILE NUMBER 2 IS SHOWN BELOW

NAME: Debo Engineering
DIVISION: Bill the Cat
FIRST INTERVIEW DATE 02-20-85
LAST INTERVIEW DATE 03-15-85
INTERVIEWS WERE PREVIOUSLY OPENED

>>> DO YOU WISH TO DELETE THE INTERVIEW FILES FOR THIS EMPLOYER? (Y/N)

Figure 23: UPDAY--Employer Information

If it is time to open interviews it will instead ask "do you wish to open interviews? (Y/N)." After opening interviews or deleting files the program automatically continues on until all files have been checked, signaling that it is done and allowing return to the EXECUTIV program.

When asked to open interview files, UPDAY creates one EMPLSCHE file of 512 bytes for each day of interviewing. The other files used are EMPLYRS and EMPLCROS, which were discussed in detail in Chapter 3. For a complete listing of UPDAY, see Appendix D.
CHAPTER 6
FUNCTIONS FOR PLACEMENT CENTER

In addition to keeping up the employer database as discussed in Chapter 5, the Placement Center staff will need to maintain the student files using STDNUP, to produce schedules of interviews using DAYSCHED, to create messages which will be printed with a student's schedule confirmation using MSGCREA, and to write a letter to any student who misses their interview schedule using DINGLTR. The use of these four programs is discussed in this chapter.

Use of STDNUP Program

In use, STDNUP is very similar to EMPLUP. They both start with menu (Figure 24), and have two pages of forms to complete (Figures 25 and 26).

Figure 24: STDNUP--Menu

This section of the program edits or creates new student files.

Do you wish to:
  1) Edit/Delete an existing student file?
  2) Create a new student file?
  3) Return to the main menu?
Choose (1, 2, or 3) please =>
Please enter information requested within the appropriate field. You should use <return> to move between the fields. Or, you may enter Menu & <return> to discard information and return to the last menu. You must not use commas in the data fields.

<table>
<thead>
<tr>
<th>STUDENT NAME</th>
<th>(LAST)</th>
<th>(FIRST)</th>
<th>(MIDDLE INITIAL)</th>
<th>SOCIAL SECURITY NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TELEPHONE NUMBER</th>
<th>(AT HOME)</th>
<th>(ANOTHER)</th>
<th>THE SECOND IS: AT WORK = 1 OR AT PARENTS = 2 OR AT FRIENDS = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCAL ADDRESS</th>
<th>(STREET)</th>
<th>(CITY)</th>
<th>(STATE)</th>
<th>(ZIP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PERMANENT ADDRESS</th>
<th>(STREET)</th>
<th>(CITY)</th>
<th>(STATE)</th>
<th>(ZIP)</th>
<th>(COUNTRY)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 25: STDNUP--First Form

Note that on the first form, the second telephone number given may be a work telephone, the parent's telephone, or at a friend's house, as indicated by the one-digit descriptor following the number. On the second form, term, year and major codes are used identically to that described in EMPLUP, and the "interview history" section is not implemented at the present time.
If this were a file which had already been created and was being updated or checked, the data would appear on the form, exactly as if it were a paper form kept in a filing cabinet.

When the student file has been created and is filed, the message shown in Figure 27 gives the file number.

Keeping the student file number, however, is not necessary as the student file is easily accessed through the social security number.

PLEASE ENTER INFORMATION REQUESTED WITHIN THE APPROPRIATE FIELD. YOU SHOULD USE <RETURN> TO MOVE BETWEEN THE FIELDS. OR, YOU MAY ENTER MENU5 & <RETURN> TO DISCARD INFORMATION AND RETURN TO THE LAST MENU. YOU MUST NOT USE COMMAS IN THE DATA FIELDS.

<table>
<thead>
<tr>
<th>BRAD DATE--TERM</th>
<th>YEAR</th>
<th>MAJOR CODE</th>
<th>-INTERVIEW HISTORY-</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) FALL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2) SPR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) SUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CITIZENSHIP CODE</td>
<td></td>
<td>DEGREE CODE</td>
<td></td>
</tr>
<tr>
<td>1) U.S.A.</td>
<td></td>
<td>1) BACH.</td>
<td></td>
</tr>
<tr>
<td>2) PERM VISA</td>
<td></td>
<td>2) MASTER</td>
<td></td>
</tr>
<tr>
<td>3) STDNT VISA</td>
<td></td>
<td>3) DOCTOR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4) CERTF.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BRADE POINT--AS OF</th>
<th>(X.XX)</th>
<th>TERM YEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 26: STDNUP--Second Form
FILING THE STUDENT DATA

STUDENT NUMBER  WAS GIVEN FILE NUMBER 4

Figure 27: STDNUP--Student File Number

STDNUP uses the STUDENT, STDNCROS, PAGEONE, and PAGETWO files, which were described in Chapter 3. For a complete listing of STDNUP, see Appendix C.

Use of DAYSCHED Program

DAYSCHED can be used to print two different types of schedules. The first is a schedule of all the interviews scheduled on a given date. The second type of schedule is one showing all the interviews for a given employer on up to all five possible interview dates--see Figure 28.

Figure 28: DAYSCHED--Menu
If the first function is selected, the program will ask for the desired day's date, check the date to make sure it is entered correctly, and start printing. The output is in the form shown in Figure 29. It should be recognized that this is quite an abbreviated example. Note the amount of information given on the student scheduled for interview.

---

**DATE:** 03-01-85

**INTERVIEWER NO. 1**

---

**TIME:** 1030 (AM)

**STUDENT NAME:** Debus, Kurt L.  **SOCIAL SECURITY:** 100-00-1111

**LOCAL TELEPHONE:** 123-000-0000  **EXPECTED GRADUATION:** SPRING TERM OF 1985

**MAJOR CODE:** 0920  **CITIZENSHIP:** PERMANENT VISA  **DEGREE:** BACHELORS


---

**INTERVIEWER NO. 2**

---

**NO INTERVIEWS SCHEDULED**

---

Figure 29: DAYSCHED--Printout

If, instead, the interviews of a particular company are desired, the interchange with the program might look like Figure 30. Here the program determines that the user does know the company's file number, the number being given
as two, the program confirms the name of the employer, and gives the dates for which interviews may have been scheduled. The print of a day's schedule in this case would be in the same format as was shown in Figure 29.

DO YOU KNOW THE EMPLOYER'S FILE NUMBER? (Y/N)

ENTER THE FILE NUMBER, THEN <CR> PLEASE===>2

THE SELECTED COMPANY IS SHOWN BELOW

COMPANY: Debo Engineering
DIVISION: Bill the Cat

IS IT CORRECT? (Y/N)

INTERVIEWS ARE SCHEDULED ON THESE DATES:

1) 02-20-85
2) 03-01-85
3) 03-15-85

DO YOU WISH A PRINT FOR ALL OF THE DATES ABOVE? (Y/N)

Figure 30: DAYSCHED--Second Option

DAYSCHED uses the EMPLYRS, EMPLCROS, and EMPLSCH file. These files were described in Chapter 3. For a complete listing of DAYSCHED, see Appendix E.
Use of DINGLTR Program

In order to use DINGLTR, the staff person must know the date the interview was missed, the name of the company, and the file number of the student who missed the interview. The staff members's interaction with the program on the screen might look like Figure 31.

**This program will write a letter to a student who has missed a scheduled interview. Please enter the information requested exactly as you wish it to appear. Do not use any commas in the data you enter!**

- **The date the student missed (like this: 2-12-85)**
  - Enter date, then <CR> ==> 3-14-85

- **The name of the interviewing company**
  - Company, then <CR> ==> Generous Electric

- **The student's system file number (1 to 4 digits)**
  - File number, then <CR> ==> 2

Figure 31: DINGLTR--Getting Started

The operator would then be asked to insert departmental stationery into the printer and line it up for printing. The letter produced appears in Figure 32.

DINGLTR uses the student's file described in Chapter 3. For a complete listing of DINGLTR, refer to Appendix F.
03-16-1985

Eduardo Contrera
007 Bond Street
Jamestown
FA 33333

Dear Student

Our records indicate that you failed to fulfill your interview appointment on 3-14-85 for your interview with Generous Electric.

If you find that unavoidable circumstances prevent you from keeping your appointment, please extend the courtesy of notifying the placement center.

Interview appointments must be fulfilled as scheduled or cancelled in advance to retain your interviewing privilege. If you fail to fulfill two interview appointments, it will be necessary to talk to a Placement Counselor prior to scheduling additional interviews.

It is important that you follow these procedures so that we can be of maximum assistance to both you and the employers.

Should you have any questions, or wish to discuss other services provided by the Placement Center, please come in to see us.

Sincerely,

James W. Bracey
Director
Career Planning & Placement

cc: department chairman

FIGURE 32: DINGLTR--Sample Letter
Use of MSGCREA Program

The fourth program created to fulfill Placement Center functions is MSGCREA, which is designed to record messages which either the staff or employer wish to give the student along with confirmation of their interview. Sample messages were shown in Figure 15 of Chapter 4.

This program edits and creates messages to the interviewer. You will be asked for the message number first, then if such a message exists, it will be shown to you—if no such message already exists, you may enter it.

Are you ready to continue? (Y/N)

Enter the message number followed by <CR> ••>3

Figure 33: MSGCREA--Starting Up

In Figure 33, the staff person has indicated that they wish to continue, and given the message number to be edited as three. If a new message were to be written, a higher (unused) message number could have been chosen, or the old message number three could have been written over. If a message with that number has already been recorded, it will appear as shown in Figure 34.
MESSAGE NUMBER 2 APPEARS BELOW, YOU MAY EDIT IT AS YOU WISH. PRESS <RETURN> WHEN YOU ARE DONE.

Please be advised that this is a long boring message to tell you that you must do or not do something that you wished you could not do or do but knew all along that you would or would not do it anyway. Right?

Figure 34: MSGCREA--Message Editing

Note, that in editing messages in MSGCREA, all the usual editing functions like insert and delete are done under control of MSGCREA. This was necessary due to the nature of the Basic language. These functions operate almost identically to Basic, however, and are easily learned by practicing.

MSGCREA used only one file, MESSAGES, which was described in Chapter 3. MSGCREA is listed in Appendix F.
CHAPTER 7

MODIFICATION AND TROUBLE SHOOTING

The system described in this paper is designed for readily updating, modifying, and trouble shooting. All of its important functions are in separate programs so that they can be worked on separately. The tasks which each of these modules perform have been discussed in the last four chapters. The software files have been designed with future growth in mind. Space has been provided for data which may have future use, and some free space exists in all files. In addition to adding totally new functions for the software to perform, it is anticipated that improvements will be made in the interface between user and program and changes will occur in the use and maintenance of the system database (files).

The functional description and extensive commenting of the programs given in the appendices will prove useful in both areas mentioned, and the FILETEST program will be especially helpful in work with the database.

Use of FILETEST Program

Although FILETEST is accessible through EXECUTIV, like all the rest of the system software, it is expected that
the user will be familiar with the Basic language and the Basic random access file structure. The program is used to determine, character by character, the contents of any system file. For that purpose, it presents a numbered scale above the characters stored in the file being examined. Figure 34 shows the program's menu.

**FILE TEST**

**WHICH TYPE OF FILE DO YOU WISH TO LOOK AT?**

1) THE EMPLOYER FILE
2) THE EMPLOYER CROSS FILE
3) THE STUDENT FILE
4) THE STUDENT CROSS FILE
5) THE EMPLOYER SCHEDULE FILE

**CHOICE (1 THROUGH 5) PLEASE**

---

Figure 35: FILETEST--Menu

After the user has entered the type of file to be examined, the program will request the file number. In Figure 35 a sample employer file is displayed. Note that characters 1 through 25 correspond to the name of the company as shown in the description of the employer file (Table 4, Chapter 3). Character 51 is the first character of the employer's address and begins on line 2 in the display. On the third "page" of the file, the first four characters give the file number for the first interview
date, (file five), which can be seen from the third line of page 2 to be on 02-20-85 (characters 125 through 130). The two interviewers have four interview times open in the morning and four in the afternoon (characters five through eight, and 14-17 of page 3), but one student has signed up, reducing the number of interviewers available to one in character six. The data in the files is stored in

<table>
<thead>
<tr>
<th>EMPLOYER NUMBER 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I...........10.......20.......30.......40.......50</td>
</tr>
<tr>
<td>I...........10.......20.......30.......40.......50</td>
</tr>
<tr>
<td>1 Debo Engineering Bill the Cat</td>
</tr>
<tr>
<td>2 12345 Minor League Orlando FL32816000</td>
</tr>
<tr>
<td>3 OU.S.A. Debo John CSir</td>
</tr>
<tr>
<td>4 3053658510 03 1842853</td>
</tr>
<tr>
<td>5 854</td>
</tr>
</tbody>
</table>

- 0900 12341Right here!
- 300Gophur 202208544203018544203158544
- 209001300

| I...........10.......20.......30.......40.......50 |
| I...........10.......20.......30.......40.......50 |
| 1 52122000002222200000 62212000002222200000 722 |
| 2 22000000222220000000000000000000000000000000000 |
| 3 0000000000000000000000000000000000000000000 |
| 4 |
| 5

———– CONTINUE? (Y/N) <<<———–

Figure 36: FILETEST--Employer File

the precise order and number of characters shown in the five tables in Chapter 5. Page 1, 2, and 3 of the employer file are 205, 177, and 130 characters respectively.
In Figure 36, the employer cross file is shown. The first interview date (02-20-85) and the last interview date (03-15-85) appear in characters 51 through 62, and that the last one in character 63 indicates that interviews have been opened, as was noticed from the employer file previously. (Prior to the opening of interviews, no data is on page 3 of the employer file.) Note also that the employer cross file and employer file numbers are the same.

EMPLOYER CROSS FILE NUMBER 2

|...........10...........20...........30...........40...........50...........60...........70 |
| l + l + l + l + l + l + l + l + l + l |
| Debo Engineering Bill the Cat 0220850315851 |

_____________>>> CONTINUE? (Y/N) <<<<<<<<<<<<<<<<<<<<<<<

Figure 37: FILETEST--Employer Cross File

In Figure 37, the student file display is shown. The first character at the beginning of page 2 is the expected graduation term. A two indicates Fall. The year follows. There are 154 characters on the first page of the student file, and 102 on the second.
<table>
<thead>
<tr>
<th>STUDENT NUMBER</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>1 Contraro Eduardo</td>
<td>L10020300011122233334</td>
</tr>
<tr>
<td>2</td>
<td>4455566661007 Bond Street Jamestown</td>
</tr>
<tr>
<td>3</td>
<td>334 10 Downing St. London</td>
</tr>
<tr>
<td>4</td>
<td>d</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1 285093031300184</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

| 5 | + | + | + | + | + | + | + | + |
|----------------|

| ___________>>> CONTINUE? (Y/N) <<<<<<<<<<|

Figure 38: FILETEST-Student File

If a copy of the display is required, simultaneously press the shift key and print screen to direct the screen contents to the printer. The message at the bottom, "continue? (y/n)" serves to hold the contents until the user is done; on pressing "y" the menu is redisplayed.

In Figure 38, the student cross file is shown. This file contains the nine-character social security number and the six-character date of last use. Notice, this student's file was last accessed on 02-16-85.
Finally in Figure 39, the employer schedule file display is shown. Page 1 shows the morning schedule with two interviewers available for four time slots (four blanks are one slot). Note that the second time slot has "1" (characters 25-28 are the file number) indicating the student with file number one has scheduled that interview. Page 2 is the afternoon schedule. On page 3 it can be seen that two interviewers are scheduled for 02-20-85 with four morning and four afternoon interviews of 45 minutes each (the two at character number 10 indicates 45 minutes), and that morning interviews start at 0900 and afternoon interviews at 1300. In this case, the interviewers have a long lunch break. Characters 19 through 22 give the employer file for which this schedule was created (in this case employer file two).
A good understanding of FILETEST is necessary to the task of revising the system's files or the use of the files. A complete listing of FILETEST is in Appendix G.

**Use of CREAPG Programs**

In order to modify the way a user interfaces with either EMPLUP or STDNUP, it is necessary to understand how the four CREAPG programs create the forms used in communicating with the two programs. The programs are listed in Appendix G, but some discussion of their use is appropriate here.
The screen of the IBM PC is divided into picture elements--25 vertically and 80 horizontally. These picture elements can be letters, numbers, blank spaces, single and double lines, and more complex shapes. The first task of the CREAPG programs is to define all of the characters needed to create the form used to communicate with the user. An example of a character might be the word "local" used in asking for a local address, or a section of line used in creating a blank to be filled in. After a table of all these elements is created, a second table is needed in order to tell the program how to draw the page. It consists of the row and column to start printing at, the number of the character defined in the table above, and the number of repetitions of that character required. The third table required tells the program where to pick up data from the form after it has been filled out or where to write the data onto the form from a data file. This table consists of the row and column location of the first data element, the number of elements to get, and any column to skip over (dashes or other division marks) in the process.

These three tables (characters, drawing and location information) are stored in the files called PAGEONE, PAGETWO, PAGETHRE, and PAGEFOUR by CREAPG1, CREAPG2,
CREAPG3, and CREAPG4. A careful study of these programs in the Appendix will allow one to make any necessary changes to the forms drawn or to create a form for a totally new function.

**Use of AUTOEXEC.BAT Program**

In order that the computer automatically loads and begins execution of software upon POWER UP, an AUTOEXEC.BAT file must exist on the primary drive disk. In the system configuration described here, the program called by this file is called STARTUP which obtains the password and allows access to EXECUTIV, which is the system's main menu, giving access to all the rest of the system software.

Within the comments of STARTUP a description of the method used in creation of the AUTOEXEC.BAT file is given. The program is easily created. Operating in the Disk Operating System, the user types "COPY CON;AUTOEXEC.BAT" and then a carriage return. The next line is optional, but it reduces screen clutter: "ECHO OFF." The third line for this system is "BASICA STARTUP.BAS/F:3/S:512." This tells the Disk Operating System that on POWER UP it is to load the version of Basic called advanced which allows three files to be open at a time, each of up to 512 characters, and then to load and run a basic program called STARTUP.
Where the program name STARTUP is used any other could be substituted. Finally, in order to create the file, press function key number six and return at the same time. It may be desirable to use a compiled Basic program without BASICA being loaded in order to increase the security of the system.

**Suggestions**

Ordinary Basic language troubleshooting techniques include the use of the trace option, which automatically gives the line number of each instruction worked. It is a good idea to put the commands TRON and TROFF within the body of the program, so that a limited amount of material is displayed on the screen. It is also very useful to use the STOP command within a program to halt it so that the values of variables, etc., may be looked at. The program may then be resumed by typing "CONT" at the keyboard. Both of these techniques are useful for regular Basic, but not after it has been compiled.

It is very useful to have the programs being modified or tested on separate floppy disks with a few test files in order that no danger of damage to the system files or other software exists. In order to test the opening and closing
of interview files, it is handy to use the Basic "DATE$" command to change the date which the software considers today. Program testing is best done on a subroutine-by-subroutine basis. When testing any program that uses files, it is necessary to know what is in the file. A printer is important here, because a hard copy of the file aids this work a great deal. It is better to reread the list of variables, the subroutine description, and the embedded remarks in the body of the code, rather than to rely on one's memory, when doing program design or trouble shooting.

This system of software has been designed in a linearly programmed, modular, heavily commented style in Basic, so that it would be easily maintained and modified. If additional work is done on the software, it is strongly suggested that the same goals be adopted in consideration of future programmers.
CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

It is felt that the system described in this paper demonstrates that microcomputer based, user friendly, and easily updated software can be written in Basic to accomplish very specialized office applications. Whether this software can be easily used by non-computer literate office personnel and maintained by student assistants with a moderate level of programming ability remains to be seen. It is fervently hoped that an opportunity to demonstrate this second point will be available.

It is intended that the system be used in parallel with the existing paperwork system during an initial, perhaps two to three month break-in period, during which any problems could be corrected and obvious needs and embellishments would be implemented. It then would be compiled for speed of operation and greater security.

In general the software developed attempts to demonstrate a philosophical decision; that the programmer should strive to design clear, cleanly written, and heavily documented software which is easily understood, easily modified, and easily used. There is no place in a
good program for the cute "tricks" and deliberate obfuscation used in so much of the code generated by programmers who seem to want to keep computer use for only the computer specialist.

One of the improvements which would augment the function of this system would be to add a program which would automatically create a list of persons to notify and telephone numbers in the event of a change in an interviewer's schedule after interviews have already been opened. Under the present system, this is less easily accomplished. Another possible improvement would be to allow the student to change more items in his or her data file without requiring staff intervention. A program to aid creation of backup copies of system files would be a useful addition. As a final software recommendation, a program which collects statistics on interviews and perhaps records interview performance in the student's file could be written.

In terms of hardware, in addition to the optional items listed in Table 1 of Chapter 3 (color monitor and clock-calendar card), a non-interruptable power supply would be an eventual positive addition, as it would improve the safety of the system's database.

When the system goes into regular use, it is suggested that daily back-up copies of the system files be kept, and
that a hard copy of interviewer schedules be printed on a regular basis. Students using the system should leave their name and telephone number on a form by the computer in the event that data is lost prior to the back-up copy being made. The machine should be located within sight of a staff person, and perhaps some sort of a cover could be used to keep a student from inserting his or her own disk.
APPENDIX A

STDNSRV--THE STUDENT SERVICE PROGRAM

This appendix presents a functional description and a complete listing of STDNSRV. The functional description is intended to be somewhat similar to a flow chart, except that it is more compact and gives the function of blocks or sections of code, rather than line by line as in a flow chart.

The use of STDNSRV is described in Chapter 4. It would be good to review that chapter before attempting to analyze the program's function.

Functional Description

STDNSRV performs four main tasks: It checks and updates those things in the student file (STUDENTS) which the student can change. It then displays some important facts in the file which the student cannot change through the program. Next, it will go through all the employers who have open interviews, check with the student, and allow the student to schedule an interview if he or she so desires. Finally the program will present a list of employers coming in the next 30 days, whether their interviews have been opened for scheduling or not.
The program comments commence with a list of important variables, and then the size of the important files is set. These file dimensions occur at the start of all the software in this system, and must be adjusted for the size of the storage medium available. For the current system, it is assumed that there will be no more than 100 student, employer, or interview files, and no more than 50 interviews planned within the next 30 days.

At line 340, subroutine "Strike any key" is called. The subroutine displays the message at random locations on the screen to avoid burning the message into the phosphor. The message is in color on a color monitor.

Lines 350 to 440 display the opening message, then subroutine "Yesno" is called. This subroutine expects a letter answer to the question asked and returns a one for yes and a two for a no. Like "Choice," this subroutine in STDNSRV only, has a time-out function. If the student leaves the console, a minute or so will elapse, then "Timeout" is called, which terminates the program.

Lines 500 to 600 obtain the student's social security number, which is redisplayed by line 630 for confirmation. If "Yesno" confirms that the number was correct, lines 740 to 810 look for a student cross file with the given social security number, get the file number for that file, and
update the use date in the cross file, showing that it is a
file which has been used recently.

Lines 910 and 960 get the student file which has the
same file number obtained above. The two pages of
information (INFO$(1 and 2)) correspond to the two form
pages used in STDNUP. (See Table 2 in Chapter 3 for
details). Also, whenever possible, a file is accessed and
then immediately closed. If a file is opened within a
subroutine, it will always be closed within the subroutine.

Lines 990 to 1100 put the student's name on the screen
and obtain confirmation that the correct file has been
accessed.

Lines 1260 and 1340 put the student's telephone number
on the screen using a data statement to give the location
of the number in INFO$(1), and subroutine "Infoget" to pull
the correct characters out of the array. Note that
"Infoget" is called with FIRSTCHAR equal to the first
character number in INFO$, and NOCHAR equal to the number
of characters to be obtained. Upon return, INFOGET$ contains the character string obtained, and the length of the string is in "L." Note, that trailing blanks are
eliminated and that if the number were all blanks, "L"
would be zero. If the telephone number in the file is
incorrect, the correct number is entered in lines 1380-1410 and put into INFO$ in line 1420 for eventual return to the student's file.

Lines 1430 and 1790 perform the identical tasks for the student's address, and in lines 1800 to 1850, the file is brought up-to-date, completing the software's first main task.

In line 1970, the last two digits of the year of the student's expected graduation are obtained and converted from characters to a number in line 1980. The graduation term is obtained as a one-digit descriptor in line 1990 and converted to the correct verbal description in lines 2000-2030. The term and year are then placed on the screen by line 2040. Similar logic to this is used to place the major code, citizenship, degree, and grade point average in lines 2060 to 2380. This completes the second main task of the program.

In lines 2450 to 2550 the program determines whether the student wishes to schedule interviews and if not, whether the student wishes a list of companies expected in the next month. If the student does schedule interviews, execution of the program will return here to allow the second choice later.

Assuming the student chose to schedule interviews, the next line (2610) calls the subroutine "Student's
qualifications," which obtains the DEGREE$, MAJOR$, GPA, CITZSHIP, and GRADDATE$ from INFO$(2). Line 2620 sets the variable OPENTINTV to one, which tells subroutine "Who's interviewing," called in the next line, that a list of companies with open interviews is desired. (Rather than a list of companies which will be coming within 30 days, regardless of whether interviews have been opened). The subroutine produces array INTFILE(INDEX), containing the employers' file numbers, by checking the employer cross files. INDEX will be the number of companies found.

In line 2650, INDEX is checked to see if any files were found. If they were, lines 2660-2715 get the three pages of data contained within the last employer file on the list. (INDEX counts down from its initial value to zero.) Lines 2740 to 2780 then print the name and division for the company. Lines 2790 to 2820 test the degree being obtained by the student to see if it is acceptable to the employer. Lines 2830 to 2930 use the same logic to check the major, citizenship, grade point, and graduation date. If these values check, the message in 2940-2950 is printed, but if not the messages in lines 3000-3070 inform the student on which item he or she was not qualified to interview with that particular company.
Assuming the student was qualified, lines 3200-3250 are executed producing the heading for the schedule listing. The outer loop beginning in line 3260, produces the date and dividing lines for the interview listing, and the inner loops using the variable "P" starting at 3450 and 3580 produce the interview times on a particular date. The interview date is THISDATE while TODAY indicates the current date. Line 3370 makes sure that dates in the past do not produce any interview times, so that they won't be inadvertently scheduled. Lines 3390-3440 get the morning start time and length of the interviews, so that lines 3510 and 3520 can calculate the time for each interview. The interview schedule format is shown in Figure 14 of Chapter 4. Lines 3470 and 3480 serve to delete the time listing if no interviews are available at that time. The afternoon schedule is produced identically to the morning schedule.

Line 3740 asks which day the student wishes to schedule, and subroutine "Choice" is used to get the number. Note, that since the highest number is five, CHOICE is set equal to five before calling the subroutine. If a number less than one or greater than five is selected, the subroutine will produce the error one and continue looking for a correct response. The logic in lines 3780 through 3910 is essentially the same as that used in finding the desired day.
Line 3950 checks to be sure that the interview time chosen was available, and if it was, line 4030 updates the interview schedule information contained in the employer file, which is then refiled in lines 4040-4080.

The actual schedule file is opened in line 4140, the correct interview time is checked in lines 4230 to 4300, and the student file number is put into the correct time slot in lines 4340 to 4400. If there was no time available, a very serious discrepancy between the information contained in the employer file and that in the employer's interview file exists. A programmer would probably be required to determine the source of the error.

The program instructs the student to be sure the printer is ready for a copy of his or her schedule (lines 4450-4470), and prints the student's name, social security number, employer name and division in lines 4500-4550. The date and time are printed using similar logic to that previously described, and any message numbers are obtained in lines 4710 and 4760-4770. The messages are printed and the student is asked whether another interview is desired. If not, the program has completed its third main task.

The fourth and last task is rather simply accomplished. With OEPNINTV set to zero, subroutine "Who's interviewing?" is called, and a list of all companies expected within the next 30 days is produced in INTFILE.
The employer file is opened in line 4950, and data from it is used to print the name, division, and dates for the company expected to be interviewing (lines 5040-5130). When the counter INDEX reaches zero, the list is complete and the program's job is done.

Please note that lines 6700-6810 serve the purpose of trapping printer errors so that the program does not terminate on a student if the printer is not turned on. It will just hold at the point where printing needs to start and continue to give the message "please check the printer."
STDNSRV.BAS Listing

LIST 10 REM STDNSRV.BAS WRITTEN 12-10-84 JOHN C. DEBO
20 REM AND 1-2, 6, 13, 19 & 20 OF 1985
30 REM TROUBLESHOOTING 2-2 & 3 OF 1985
40 REM
50 REM
60 REM
70 REM
80 REM STDNSRV IS USED BY THE STUDENT IN ORDER TO SIGN UP
90 REM FOR INTERVIEWS. IT ALSO ALLOWS THE STUDENT TO
100 REM UPDATE CERTAIN INFORMATION IN THE STUDENT'S FILE,
110 REM AND IT WILL GIVE THE STUDENT A LIST OF COMPANIES
120 REM INTERVIEWING DURING THE NEXT MONTH.
130 REM
140 REM INFO(1 THRU 6) IS USED TO HOLD EMPLOYER AND
150 REM STUDENT FILE INFORMATION AND TO GET INTERVIEW
160 REM FILE INFORMATION ALSO.
170 REM
180 REM MAXSTU, MAXEMP, AND MAXINTV CONTAIN THE ALLOWABLE
190 REM SIZE FOR THE STUDENT, EMPLOYER, AND INTERVIEW FILES.
200 REM THESE MUST BE SET TO REFLECT THE SIZE OF STORAGE
210 REM MEDIUM AVAILABLE.
220 REM
230 REM CROSREFS CONTAINS THE DATA FROM THE EMPLOYER
240 REM CROSS FILE POINTED TO BY FILENO.
250 REM
260 REM ROW AND COL ALWAYS REFER TO SCREEN LOCATIONS.
270 REM
280 REM YESNO AND CHOICE ARE EXPLAINED IN THEIR SUBROUTINES.
290 REM
300 MAXSTU = 100
310 MAXEMP = 100
320 MAXINTV = 100
330 DIM INFFILE(50) 'MAXIMUM NO. LISTABLE INTERVIEWS
340 GOSUB 5600
350 CLS: LOCATE 5, 1
360 PRINT" WELCOME TO THE UNIVERSITY OF CENTRAL FLORIDA'S PLACEMENT CENTER"
370 PRINT" :- STRING(63, 205)"
380 PRINT" I'M HERE TO AID YOU IN SIGNING UP FOR PLACEMENT INTERVIEWS."
390 PRINT" I'LL ALSO HELP YOU TO UPDATE YOUR FILE -- ADDRESS, TELEPHONE "
400 PRINT" NUMBER, ETC.
410 PRINT" NUMBER, ETC.
420 PRINT"
430 PRINT"
440 PRINT" ARE YOU READY TO CONTINUE? (Y OR N)"
450 GOSUB 5200
460 ON YESNO GOTO 490, 350
470 REM ------------------------------------------
480 REM SECTION WHICH LOCATES THE STUDENT'S FILE
490 CLS: LOCATE 5, 1
500 PRINT" GOOD! FIRST I'LL NEED YOUR SOCIAL SECURITY NUMBER SO I CAN LOOK "
510 PRINT" UP YOUR FILE. AFTER THAT, I'LL GIVE YOU AN OPPORTUNITY TO MAKE"
PRINT" CORRECTIONS IF YOU WOULD LIKE TO."

PRINT "PLEASE ENTER YOUR SOCIAL SECURITY NUMBER EXACTLY LIKE THE ONE"

PRINT "I'VE SHOWN HERE:  123-45-6789 --THEN PRESS RETURN (";RETNo;")"

PRINT "YOUR NUMBER ==> "

SOUND 523,3

INPUT ";,N#

SSNCODE$ = LEFT$(N#,3) + MID$(N#,5,2) + MID$(N#,8)

PRINT "JUST TO BE SURE I'VE GOT IT RIGHT, DID YOU GIVE ME SOCIAL SECURITY"

NUMBER ";LEFT$(SSNCODE$,3);"+-"MID$(SSNCODE$,4,2);"+-"MID$(SSNCO

de$,6);"? (Y OR N)"

GOSUB 5200

ON YESNO GOTO 710,660

CLS LOCATE 5,1

PRINT "LET'S TRY IT OVER AGAIN. IT'S IMPORTANT THAT YOU START THE NUMB
ER";

PRINT "RIGHT BELOW THE SAMPLE NUMBER, AND THAT YOU LEAVE A SPACE, OR PU
T"

PRINT "A DASH WHERE IT'S SHOWN IN THE SAMPLE BELOW."

GOTO 530

PRINT "ALRIGHT...NOW I NEED TO LOOK FOR YOUR FILE."

PRINT "JUST A MOMENT, PLEASE."

OPEN "R", 1, "B:STDNCRROS", 16

FIELD 1, 16 AS CROSREF$;

GET 1

IF LOC(1) = MAXSTU + 1 THEN 830

IF CROSREF$ = STRING$(16,0) THEN 830

IF LEFT$(CROSREF$,9) <> SSNCODE$ THEN 760

FILENO = LOC(1)

USEDATE$ = LEFT$(DATE$,2) + MID$(DATE$,4,2) + MID$(DATE$,9)

LSET CROSREF$ = SSNCODE$ + USEDATE$

PUT 1,FILENO

CLOSE 1

RETRY

PRINT "I'M SORRY---I CANNOT FIND YOUR SOCIAL SECURITY NUMBER LISTED."

PRINT "DO YOU WISH TO RE-ENTER IT AND TRY AGAIN? (Y OR N)"

GOSUB 5200

ON YESNO GOTO 660,350

-----

SECTION WHICH GETS STUDENTS DATA FROM FILE

OPEN "R", 1, "B:STUDENTS", 256

FIELD 1, 154 AS PG1INFO$, 102 AS PG2INFO$

GET 1, FILENO

CLOSE 1

INFO$(1) = PG1INFO$

INFO$(2) = PG2INFO$

NOTE THAT INFO$(#) CONTAINS STUDENT DATA NOW.

FIRSTCHAR = 1:NOCHAR = 15:INFO$ = 1

GOSUB 5830

LASTNAME$ = INFOGET$

FIRSTCHAR = 16:NOCHAR = 14

GOSUB 5830

FIRSTNAME$ = INFOGET$

MIDINIT$ = MID$(INFO$(1),30,1)

CLS: LOCATE 2,1
PRINT" FOR YOUR SOCIAL SECURITY NUMBER (";LEFT$(SSNCODE:,3);"-";MID$(SSNCODE:,4,2);"-";MID$(SSNCODE:,6,4)); MY FILES SHOW THIS NAME:

PRINT" FIRSTNAME;" MIDINIT;" LASTNAME;" ---IS THAT YOUR NAME? (Y OR N)

GOSUB 5200

ON YESNO GOTO 1170,1110

PRINT" PERHAPS YOU ARE NOT YET IN MY FILES. THERE ARE TWO THINGS YOU

CAN DO--TRY A DIFFERENT SOCIAL SECURITY NUMBER OR ASK A STAFF PERSON"

GOSUB 5200

ON YESNO GOTO 1170,1110

CLS: PRINT

SECTION WHICH ALLOWS STUDENT TO UPDATE HIS

ADDRESS AND PHONE NUMBER.

PRINT" GOOD! NOW WE CAN GET DOWN TO BUSINESS. FIRST, "FIRSTNAME;

PRINT" LETS CHECK SOME OF THE INFORMATION IN YOUR FILE. "

PRINT" FOR ANY ITEM THAT'S WRONG TYPE 'N'. TYPE THE RIGHT"

PRINT" INFORMATION BELOW IT. THEN PRESS RETURN ( "RETN;" )"

PRINT STRING$(79,196)

LOCATE 8,5

DATA 40,3,43,3,46,4

PRINT"YOUR HOME (LOCAL) TELEPHONE NUMBER--"

LOCATE 10,10

FOR N = 1 TO 3

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT"-"INFOGET;"-

PRINT STRING$(29,-)"

LOCATE 11,10

PRINT"... IS THIS NUMBER CORRECT? (Y OR N)

GOSUB 5200

ON YESNO GOTO 1440,1380

LOCATE 11,10

PRINT"...-"INFOGET;"

FOR N = 1 TO 3

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT" "INFOGET;

NEXT N

PRINT STRING$(1,29);" ...DO I HAVE YOUR ADDRESS RIGHT? (Y OR N)

GOSUB 5200

ON YESNO GOTO 1800,1560

LOCATE 11,10

PRINT" "INFOGET;

PRINT" "INFOGET;

PRINT"...-"INFOGET;

FOR N = 1 TO 4

LOCATE 9 + 2*N, 10

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT" "INFOGET;

PRINT"

GOSUB 5200

ON YESNO GOTO 5400,4900

LOCATE 8,5

DATA 61,20,81,15,96,2,98,5

PRINT"YOUR LOCAL ADDRESS (STREET/CITY/STATE/ZIP)

FOR N = 1 TO 4

LOCATE 8 + 2*N, 10

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT" "INFOGET;

NEXT N

PRINT"...DO I HAVE YOUR ADDRESS RIGHT? (Y OR N)

GOSUB 5200

ON YESNO GOTO 1800,1560

LOCATE 11,10

PRINT" "INFOGET;

PRINT" "INFOGET;

PRINT"...-"INFOGET;

FOR N = 1 TO 4

LOCATE 9 + 2*N, 10

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT" "INFOGET;

PRINT"

GOSUB 5200

ON YESNO GOTO 5400,4900

LOCATE 8,5

DATA 61,20,81,15,96,2,98,5

PRINT"YOUR LOCAL ADDRESS (STREET/CITY/STATE/ZIP)

FOR N = 1 TO 4

LOCATE 8 + 2*N, 10

READ FIRSTCHAR,NOCHAR

GOSUB 5830

PRINT" "INFOGET;

PRINT"...DO I HAVE YOUR ADDRESS RIGHT? (Y OR N)

GOSUB 5200

ON YESNO GOTO 5400,4900
1650 INPUT "" , N$
1660 IF LEN(N$) < 1 THEN 1680
1670 MIDS(INFO$ (1) , 81 , 15) = LEFT$( (LEFT$(N$, 15) + SPACE$( 15 ) ) , 15)
1680 LOCATE 15 , 10
1690 PRINT "... (STATE--FLORIDA IS 'FL')"
1700 LOCATE 15 , 10
1710 INPUT "" , N$
1720 IF LEN(N$) < 1 THEN 1740
1730 MIDS(INFO$ (1) , 96 , 2) = LEFT$( (LEFT$(N$, 2) + SPACE$( 2 ) ) , 2)
1740 LOCATE 17 , 10
1750 PRINT "...... (ZIP CODE--5 FIGURES)
1760 LOCATE 17 , 10
1770 INPUT "" , N$
1780 IF LEN(N$) < 1 THEN 1800
1790 MIDS(INFO$ (1) , 98 , 5) = LEFT$( (LEFT$(N$, 5) + SPACE$( 5 ) ) , 5)
1800 OPEN "R", 1 , "B:STUDENTS", 256
1810 FIELD 1 , 154 AS PG1INFO$, 102 AS PG2INFO$
1820 LSET PG1INFO$ = INFO$( 1 )
1830 LSET PG2INFO$ = INFO$( 2 )
1840 PUT 1 , FILENO
1850 CLOSE 1
1860 REM --------------------------------------------------
1870 REM THIS SECTION DISPLAYS THE STUDENT'S G.P.A.,
1880 REM GRADUATION DATE, MAJOR, AND CITIZENSHIP.
1890 REM TO CHANGE THESE, A FORM MUST BE FILLED OUT.
1900 CLS
1910 LOCATE 2 , 1
1920 PRINT "SOME OTHER INFORMATION WHICH I HAVE ON FILE IS SHOWN BELOW."
1930 PRINT "IN ORDER TO MAKE ANY CHANGES IN IT YOU WILL NEED TO FILL OUT"
1940 PRINT "A FORM WHICH YOU WILL FIND NEAR THIS CONSOLE. (A STAFF"
1950 PRINT "PERSON MUST VERIFY THE INFORMATION AND PUT IT ON...
1960 PRINT "..."
1970 PRINT "RETURN TO IT AGAINST"
1980 PRINT "LIST OF MAJORS IN THE SCHOOL CATALOG OR AGAINST A LIST"
1990 PRINT "...MAY BE FOUND NEAR THIS CONSOLE."
2000 PRINT "ARE YOU READY TO CONTINUE? (Y OR N)
2010 GOSUB 5200
2020 ON YESNO GOTO 2130 , 2110
2130 LOCATE CSLIN -1 , 1
2140 N$ = MIDS(INFO$ (2) , 8 , 1)
2150 CITIZ$ = "NOT HAVING GIVEN US YOUR CITIZENSHIP STATUS."
2160 IF N$ = "1" THEN CITIZ$ = "BEING A US CITIZEN."
2170 IF N$ = "2" THEN CITIZ$ = "HAVING A PERMANENT VISA."
2180 IF N$ = "3" THEN CITIZ$ = "HAVING ONLY A WORK PERMIT."
2190 PRINT "I SHOW YOU AS " CITIZ$;PRINT
2200 PRINT
2210 N$ = MIDS(INFO$ (2) , 9 , 1)
2220 DEGR$ = "N UNKNOWN DEGREE."
2230 IF N$ = "1" THEN DEGR$ = "BACHELORS DEGREE."
2240 IF N$ = "2" THEN DEGR$ = "MASTERS DEGREE."
2250 IF N$ = "3" THEN DEGR$ = "DOCTORATE."
2260 IF N$ = "4" THEN DEGR$ = "SPECIAL CERTIFICATE."
4) MY FILES INDICATE YOU HAVE OR ARE SEEKING A "DEGREE"
2270 PRINT
2280 PRINT
2290 YEAR$ = MID$(INFO$(2), 14, 2)
2300 IF VAL(YEAR$) < 83 THEN YEAR$ = "??"
2310 N$ = MID$(INFO$(2), 13, 1)
2320 TERM$ = "UNKNOWN"
2330 IF N$ = "1" THEN TERM$ = "FALL"
2340 IF N$ = "2" THEN TERM$ = "SPRING"
2350 IF N$ = "3" THEN TERM$ = "SUMMER"
2360 GPA$ = VAL(MID$(INFO$(2), 10, 3)) / 100
2370 PRINT "5) AND IN THE "TERM$" TERM OF "YEAR$", YOUR GRADE POINT "
2380 PRINT "AVERAGE WAS "GPA
2390 PRINT "ARE YOU READY TO CONTINUE? (Y OR N)"
2400 GOSUB 5200
2410 ON YESNO GOTO 2450, 2400
2420 REM ----------------------------------
2430 REM DOES STUDENT WISH TO SCHEDULE AN
2440 REM INTERVIEW OR GET LIST FOR NEXT MD.?
2450 CLS: LOCATE 5, 1
2460 PRINT " DO YOU WISH TO SCHEDULE AN INTERVIEW TODAY? (Y/N)"
2470 GOSUB 5200
2480 ON YESNO GOTO 2580, 2490 "FOR INTERVIEW SCHEDULING
2490 PRINT: PRINT........... DO YOU WISH A LISTING OF COMPANIES WHO ARE INTERVIEWING
DURING"
2500 PRINT " THE NEXT MONTH?"
2510 GOSUB 5200
2520 ON YESNO GOTO 4900, 2530 "FOR LISTINGS OF COMPANIES
2530 PRINT: PRINT" YOUR SESSION IS NOW COMPLETE. IT HAS BEEN A PLEASURE TO SER
2540 PRINT" PLEASE LET THE NEXT PERSON TAKE A SEAT. THANK YOU."
2550 GOSUB 5200
2560 RUN
2570 REM -------
2580 REM SECTION WHICH MATCHES OPEN INTERVIEWS
2590 REM STUDENT QUALIFICATIONS.
2600 REM
2610 GOSUB 6470 "GET STUDENT'S QUALIFICATIONS
2620 OPENINTV = 1 "GET EMPLOYERS WITH INTERVIEWS OPENED
2630 GOSUB 6180 "SIZE OF INTFILE() DETERMINED
2640 NUMINTV = INDEX "NUMINTV
2650 IF INDEX <= 0 THEN 3140
2660 OPEN "R", 3, "B:EMPLYRS", 512
2670 FIELD 3, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
2680 GET 3, INTFILE(INDEX)
2690 INFO$(1) = PG1INFO$
2700 INFO$(2) = PG2INFO$
2710 INFO$(3) = PG3INFO$
2715 CLOSE 3
2720 REM NOTE THAT INFO$(4) CONTAINS EMPLOYER DATA NOW.
2730 REM -------
2740 CLS: LOCATE 5, 1
2750 PRINT " COMPANY NAME: "MID$(INFO$(1), 1, 25)
2760 PRINT
2770 PRINT " DIVISION: "MID$(INFO$(1), 26, 25)
2780 PRINT
2790 FOR N = 41 TO 44
2800 IF MID$(INFO$(2), N, 1) = DEGREE THEN 2830
2810 NEXT N
2820 GOTO 3000 "WRONG DEGREE
2830 FOR N = 1 TO 37 STEP 4
2840 IF VAL(MID$(INFO$(2), N, 4)) = VAL(MAJOR$) OR MID$(INFO$(2), N, 4) = "0000"
2850 IF VAL(MID$(INFO$(2),N,4)) = INT(VAL(MAJOR$)/100) & 100 THEN 2880
2860 NEXT N
2870 GOTO 3010 'WRONG MAJOR
2880 IF GPA < VAL(MID$(INFO$(2),71,3)) THEN 3020 'G.P.A. TOO LOW
2890 FOR N = 194 TO 203 STEP 3
2900 IF GRADDATES = MID$(INFO$(1),N,3) THEN 2940
2910 NEXT N
2920 GOTO 3040 'WRONG GRAD. DATE
2930 PRINT "--WOULD BE INTERESTED IN TALKING TO YOU. DO YOU WISH TO"
2940 PRINT "SET UP AN INTERVIEW WITH THEM? (Y/N)"
2950 GOSUB 5200
2960 ON YESNO GOTO 3170,2980
2970 INDEX = INDEX - 1
2980 IF INDEX < 0 THEN 3140
2990 GOTO 2660
3000 PRINT "IS NOT SEEKING PERSONS WITH YOUR DEGREE.";GOTO 3050
3010 PRINT "IS NOT INTERVIEWING YOUR MAJOR.";GOTO 3050
3020 PRINT "IS NOT INTERVIEWING PERSONS WITH YOUR GRADE POINT AVERAGE."
3030 PRINT "CAN NOT EMPLOY PEOPLE WITH YOUR CITIZENSHIP STATUS.";GOTO 3050
3040 PRINT "IS NOT SEEKING PEOPLE WITH YOUR GRADUATION DATE."
3050 PRINT;PRINT "I AM SORRY. IF YOU ARE CONVINCED THAT YOU ARE QUALIFIED, YOU WILL"
3060 PRINT "NEED TO SPEAK TO A STAFF PERSON ABOUT IT. PERHAPS YOU CAN SIGN UP"
3070 PRINT "ON A WAITING LIST.";PRINT
3080 PRINT "ARE YOU READY TO CHECK THE NEXT COMPANY? (Y/N)"
3090 GOSUB 5200
3100 ON YESNO GOTO 2980,3090
3110 INDEX = INDEX - 1
3120 IF INDEX = 0 THEN 3140
3130 GOTO 2680
3140CLS:LOCATE 5,1
3150 PRINT "THERE ARE NO MORE COMPANIES INTERVIEWING AT THIS TIME."
3160 GOTO 2490
3170 REM "INTERVIEW DATE/TIMES AVAILABLE TODAY'S DATE"
3180 REM "INTERVIEW CODE 1 2 3 4 5 6 7 8 9"
3190 REM "INTERVIEW CODE 1 2 3 4 5 6 7 8 9"
3200 REM "FIRST GET THE AVAILABLE DATES AND TIMES."
3210 PRINT;PRINT "INTERVIEW DATE/TIMES AVAILABLE TODAY'S DATE"
3220 PRINT STRING$(79,196)
3230 PRINT"DAY DATE TIMES AVAILABLE ON THAT DATE (AM THEN PM)"
3240 PRINT"CODE 1 2 3 4 5 6 7 8 9"
3250 PRINT STRING$(79,196)
3260 N = 1
3270 INFONO = 2
3280 NOCHAR = 6
3290 FIRSTCHAR = N*9 + 116 'POINTS TO INTERVIEW DATE
3300 GOSUB 5900
3310 THISDATE = VAL(MID$(INFOGET$,5,2))*10000 + VAL(MID$(INFOGET$,1,2))*100 + VAL(MID$(INFOGET$,3,2))
3320 LOCATE 4+N*3,2
3330 PRINT Nl " " 'NOW PRINT THE DATE
3340 PRINT MID$(INFOGET$,1,2);"";MID$(INFOGET$,3,2);"";MID$(INFOGET$,5,2);""
3350 LOCATE 6+N*3,1
3360 PRINT STRING$(79,196) 'LINE BETWEEN INTERVIEW DATES
3370 IF THISDATE < TODAY THEN 3680
3380 REM "THAT DATE WAS PAST--GO TO NEXT DATE"
3390 INFONO = 2;NOCHAR = 4;FIRSTCHAR = 170
3400 GOSUB 5900 'GET AM INTV. START TIME
3410 INTHR = VAL(INFOGET$)
3420 NOCHAR = 1:FIRSTCHAR = 169
3430 GOSUB 5900 *GET INTV. LENGTH
3440 INTERVAL = 15 + 15*VAL(INFOGET*)
3450 FOR F = 1 TO 9
3460 INFOF = 3:NOCHAR = 1:FIRSTCHAR = 4 + P + 22*(N-1)
3470 GOSUB 5900 *ANY INTVS. AVAILABLE?
3480 IF VAL(INFOGET*) > 6 OR VAL(INFOGET*) < 1 THEN 3540
3490 REM NO AM INTERVIEWS AVAILABLE
3500 LOCATE 4+N$3,11+P$7 *TRANSLATE TIME OF INTERVIEW
3510 MIN = 60 * FIX(INTHR/100) + INTERVAL*(P-1) + INTHR - 100*FIX(INTHR/100)
3520 HRMN = FIX(MIN/60) + ( (MIN/60)-FIX(MIN/60))*60
3530 PRINT HRMN;
3540 NEXT P
3550 INFOF = 3:NOCHAR = 4:FIRSTCHAR = 174
3560 GOSUB 5900 *GET PM INTV. START TIME
3570 INTHR = VAL(INFOGET*)
3580 FOR P = 1 TO 9
3590 INFOF = 3:NOCHAR = 1:FIRSTCHAR = 13 + P + 22*(N-1)
3600 GOSUB 5900 *ANY PM INTVS. AVAILABLE?
3610 IF VAL(INFOGET*) > 6 OR VAL(INFOGET*) < 1 THEN 3670
3620 REM NO PM INTERVIEWS AVAILABLE
3630 LOCATE 5+N$3,11+P$7 *TRANSLATE TIME OF INTERVIEW
3640 MIN = 60 * FIX(INTHR/100) + INTERVAL*(P-1) + INTHR - 100*FIX(INTHR/100)
3650 HRMN = FIX(MIN/60) + ( (MIN/60)-FIX(MIN/60))*60
3660 PRINT HRMN;
3670 NEXT P
3680 N = N + 1
3690 IF N <= 5 THEN 3270 *GET THE NEXT INTERVIEW DATE
3700 REM ----------------------
3710 REM NOW ALL DATES & TIMES LISTED. FIND OUT WHICH
3720 REM INTERVIEW THE STUDENT WANTS.
3730 LOCATE 22,1
3740 PRINT "WHICH DAY DO YOU WANT? (ENTER 'DAY CODE' 1 TO 5)"
3750 CHOICE = 5
3760 GOSUB 5400
3770 DAY = CHOICE
3780 LOCATE 22,1
3790 PRINT "ON DAY ";DAY;"," DO YOU WANT A MORNING OR AFTERNOON INTERVIEW?"
3800 PRINT "(ENTER 1 FOR AM OR 2 FOR PM)"
3810 CHOICE = 2
3820 GOSUB 5400
3830 MORNAFT = CHOICE
3840 LOCATE 22,1
3850 X$ = "MORNING"
3860 IF MORNAFT = 2 THEN X$ = "AFTERNOON"
3870 PRINT "ON THE ";X$;" OF DAY ";DAY;"," WHICH INTERVIEW TIME DO YOU WISH? "
3880 PRINT "(ENTER 1 TO 9)
3890 CHOICE = 9
3900 GOSUB 5400
3910 HOUR = CHOICE
3920 LOCATE 22,1
3930 PRINT "THANK YOU. YOUR CHOICE OF HOUR NUMBER ";HOUR;" ON THE ";X$;" OF "
3940 PRINT "DAY ";DAY;"," IS BEING RECORDED."
3950 INFOF = 3:FIRSTCHAR = (DAY-1)*22 + 4 + 9*(MORNAFT-1) + HOUR:NOCHAR = 1
3960 GOSUB 5900
3970 INTVS = VAL(INFOGET*) - 1
3980 IF INTVS >= 0 THEN 4030 *THE TIME WAS AVAILABLE
3990 LOCATE 22,1
4000 PRINT "THE INTERVIEW TIME WHICH YOU CHOSE WAS NOT AVAILABLE. TRY AGAIN,"
4010 PRINT "AND BE VERY CAREFUL TO CHOOSE INTERVIEW TIMES WHICH ARE OPEN."
4015 FOR N = 1 TO 2000:NEXT N *TIME TO READ MESSAGE
4020 GOTO 3730 *WHAT DO I DO NOW?
4030 MID$(INFO$(3),FIRSTCHAR,1) = RIGHT$(STR$(INTVS),1)
4040 OPEN "R",3,"B:EMPLYS", 512
4050 FIELD 3, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
4055 LSET PG1INFO$ = INFO$(1)
4056 LSET PG2INFO$ = INFO$(2)
4057 LSET PG3INFO$ = INFO$(3)
4070 PUT 3, INTFILE(INDEX)
4080 CLOSE 3
4090 REM """"""""""""
4100 REM -EMPLYRS UPDATED, NOW PUT STUDENT IN EMPLSCHE.
4110 REM THE SCHEDULE FILE NUMBER IS IN EMPLYR (P.3).
4120 INFONO = 3:FIRSTCHAR = (DAY - 1)*22 + 1:NOCHAR = 4
4130 GOSUB 5900
4140 OPEN "R", 1, "B;EMPLSCHE", 512
4150 FIELD 1, 216 AS PG1INFO$, 216 AS PG2INFO$, 80 AS PG3INFO$
4160 FLIT = VAL(INFOGET$)
4170 GET 1, FLIT 'THE FILE NUMBER
4180 INFO$(4) = PG1INFO$
4190 INFO$(5) = PG2INFO$
4200 INFO$(6) = PG3INFO$
4210 REM NOTE THAT INFO$(4-6) CONTAIN SCHEDULE FILE DATA NOW.
4220 REM """"""""""
4230 INFONO = MORNAFT + 3 'PAGE ONE OR TWO
4240 NOCHAR = 4
4250 INTNO = 1 'INTERVIEWER NUMBER ONE FIRST
4260 FIRSTCHAR = (HOUR - 1)*24 + 1 + (INTNO - 1)*4
4270 GOSUB 5900
4280 IF L = 0 THEN 4335 '4 BLANKS SO AVAILABLE
4290 INTNO = INTNO + 1 'CHECK NEXT INTERVIEWER
4300 IF INTNO < 10 THEN 4260
4310 PRINT "****SYSTEM ERROR -- INTERVIEW FILE FULL*****"
4320 PRINT "PLEASE CALL A STAFF PERSON. JOB TERMINATED"
4330 END
4335 INFOGET$ = ""'
4340 RSET INFOGET$ = RIGHTS$(STRING$(FILENO), 4) 'STUDENT FILE NUMBER
4350 MIDS$(INFO$(INFONO), FIRSTCHAR, 4) = INFOGET$
4360 LSET PG1INFO$ = INFO$(4)
4370 LSET PG2INFO$ = INFO$(5)
4380 LSET PG3INFO$ = INFO$(6)
4390 PUT 1, FLIT
4400 CLOSE 1
4410 REM """"""""""
4420 REM FILES UPDATED WITH NEW INTERVIEW--NOW
4430 REM PRINT THE INTERVIEW SCHEDULED. USE LPRINT
4440 REM IN THE FINAL VERSION. ***
4450 CLS
4460 PRINT:PRINT" IF THE PRINTER IS OFF, TURN IT ON. IF THE 'ON LINE' LAMP IS NOT "
4470 PRINT" LIT, PRESS 'ON LINE'. ARE YOU READY TO CONTINUE? (Y/N)"
4480 GOSUB 5200
4490 ON YESNO GOTO 4500, 4480
4495 ON ERROR GOTO 6700
4500 PRINT:PRINT" INTERVIEW SCHEDULED FOR """"FIRSTNAME"""", """"MIDINIT"""", """"LASTNAME"""
4510 PRINT" SOCIAL SECURITY NUMBER """"LEFT$(SSNCODE$, 3);""""MIDS$(SSNCODE$, 4, 2); """"MIDS$(SSNCODE$, 6)
4520 PRINT:PRINT" COMPANY NAME: """"MIDS$(INFO$(1), 1, 25)
4530 PRINT:PRINT" DIVISION: """"MIDS$(INFO$(1), 26, 25)
4540 PRINT:PRINT" INFONO = 2: NOCHAR = 6; FIRSTCHAR = DAY$$ + 116
4570 GOSUB 5900 'GET INTERVIEW DATE
4580 PRINT:PRINT" DATE: """"MIDS$(INFOGET$(1), 1, 2);""""MIDS$(INFOGET$(3), 2);""""MIDS$(INFOGET$(5), 2)
4590 X$ = """"MORNING"
4600 IF MORNAFT = 2 THEN X$ = """"AFTERNOON"
4610 INFONO = 2; NOCHAR = 4; FIRSTCHAR = 170 + 4 & (MORNAFT - 1)
4620 GOSUB 5900 'GET INTERVIEW START TIME
4630 INTHR = VAL(INFOGET$)
4640 NOCHAR = 1; FIRSTCHAR = 169
4650 GOSUB 5900  "GET INTERVIEW LENGTH
4660 INTERVAL = 15 * 15 * VAL(INFOGET$)
4670 MIN = 60 * FIX(INTHR/100) + INTERVAL * (HOUR-1) + INTHR - 100 * FIX(INTHR/100)
4680 HRMN = FIX(MIN/60) * 100 + ((MIN/60)-FIX(MIN/60)) * 60
4690 PRINT; PRINT" IN THE "  & X$; " AT "  & HRMN; ". PLEASE BE ON TIME."
4700 PRINT; PRINT" THE INTERVIEWS HAVE ASKED THAT YOU BE GIVEN THE FOLLOWING INFORMATION:
4710 INFO $; 11 NOCHAR = 2
4720 OPEN "R", 1, "B:MESSAGES", 256
4730 FIELD 1, 255 AS MESSAGE$
4740 FOR N = 1 TO 5
4750 FIRSTCHAR = 184 + 2 * (N - 1)
4760 GOSUB 5900  "GET MESSAGE NUMBER
4770 MSGNO = VAL(INFOGET$)
4780 IF MSGNO < 1 OR MSGNO > 99 THEN 4820
4790 GET 1, MSGNO
4800 PRINT "MESSAGE "  & N; " IS: "MESSAGE$
4810 NEXT N
4820 CLOSE 1
4830 PRINT; PRINT" DO YOU WISH TO SCHEDULE ANOTHER INTERVIEW? (Y/N)"
4840 GOSUB 5200
4850 PRINT" IN CASE MORE INTERVIEWS
4860 PRINT " ALL INTERVIEWS HAVE ASKED THAT YOU BE GIVEN THE FOLLOWING INFORMATION:
4870 INDEX = INDEX - 1
4880 GOSUB 6180  "LIST ALL COMPANIES
4890 IF INDEX = 0 THEN 5170 "ALL DONE
4900 REM SECTION FOR LISTING MONTH'S INTERVIEWS
4910 CLS
4920 OPENINTV = 0  "LIST ALL COMPANIES
4930 GOSUB 6180
4940 NUMINTV = INDEX  "HOW MANY COMPANIES?
4950 OPEN "R", 3, "B:EMPLOYRS", 512
4960 FIELD 3, 205 AS P61INFO$, 177 AS P62INFO$, 130 AS P63INFO$
4970 IF INDEX = 0 THEN 5170 "ALL DONE
4980 GET 3, INTFILE(INDEX)
4990 INFO$(1) = P61INFO$
5000 INFO$(2) = P62INFO$
5010 INFO$(3) = P63INFO$
5020 REM NOTE THAT INFO$(4) CONTAINS EMPLOYER DATA NOW.
5030 REM THESE PRINT STATEMENTS WILL BE 'IPRINT' IN FINAL VERSION.
5040 REM -------------------
5050 ON ERROR GOTO 0
5060 PRINT" COMPANY NAME: "MID$(INFO$(1),1,25)
5070 PRINT" DIVISION: "MID$(INFO$(1),26,25)
5080 PRINT" IS EXPECTED ON THE FOLLOWING DATES... "
5090 INFON = 2: NOCHAR = 6
5100 FOR N = 1 TO 5
5110 FIRSTCHAR = N*9 + 116
5120 GOSUB 5900
5130 IF INFOGET$ = STRINGS$(6,0) OR INFOGET$ = STRINGS$(6,32) THEN 5130
5140 PRINT" "  & N; " "MID$(INFOGET$1,2);""MID$(INFOGET$,3,2);""MID$(INFOGET$,5,2)
5150 NEXT N
5160 PRINT
5170 INDEX = INDEX - 1
5180 PRINT " THAT'S ALL THE INTERVIEWERS THAT I KNOW OF. STAY IN TOUCH FOR LATER!"
5190 PRINT " INFORMATION, BECAUSE IT DOES CHANGE QUITE OFTEN.
5190 GOTO 2530 'GOODBYE
5200 REM
5210 REM SUBROUTINE YESNO
5220 REM
5230 REM THIS SUBROUTINE HANDLES YES/NO QUESTIONS, RETURNING YESNO = 1 FOR YES AND YESNO = 2 FOR NO
5240 REM
5250 REM
5260 REM
5270 SOUND 523, 3
5280 NS = ""
5290 N = 1
5300 WHILE LEN(NS) < 1
5310 NS = INKEY$
5320 N = N + 1
5330 IF N = 5000 THEN GOSUB 5970 'TIME LIMIT
5340 WEND
5350 IF NS = "N" OR NS = "n" THEN YESNO = 2:RETURN
5360 IF NS = "Y" OR NS = "y" THEN YESNO = 1:RETURN
5370 SOUND 100, 4
5380 GOTO 5280
5390 REM
5400 REM SUBROUTINE CHOICE
5410 REM
5420 REM THIS SUBROUTINE HANDLES CHOICES OF MORE THAN ONE OPTION. CALL IT WITH CHOICE = NUMBER OF LARGEST OPTION; IT RETURNS WITH CHOICE = NUMBER OF OPTION CHOSEN.
5430 REM
5440 REM
5450 REM
5460 REM
5470 REM
5480 SOUND 523, 3
5490 NS = ""
5500 M = 1
5510 WHILE LEN(NS) < 1
5520 NS = INKEY$
5530 M = M + 1
5540 IF M = 5000 THEN GOSUB 5970
5550 WEND
5560 N = FIX(VAL(NS))
5570 IF N < 1 OR N > CHOICE THEN SOUND 100, 4: GOTO 5490
5580 CHOICE = N
5590 RETURN
5600 REM
5610 REM SUBROUTINE 'STRIKE ANY KEY'
5620 REM
5630 REM PRINTS 'STRIKE ANY KEY' UNTIL A KEY IS FINALLY STRUCK. THEN STDNSRV STARTS.
5640 REM
5650 REM
5660 RANDOMIZE 111
5670 NS = ""
5680 M = 1
5690 ROW = INT(23*RND + 1)
5700 COL = INT(65*RND + 1)
5710 TINT = INT(B*RND + 1)
5720 CLB
5730 LOCATE ROW, COL
5740 COLOR TINT, 0
5750 PRINT "STRIKE ANY KEY"
5760 WHILE LEN(NS) < 1
5770 NS = INKEY$
5780 M = M + 1
5790 IF M = 500 THEN 5670
5800 WEND
5810 COLOR 6, 0
SUBROUTINE INFOGETS

THIS TAKES AN INFO$(INFONO) ARRAY, GETS
A NOCHAR (NUMBER OF CHARACTERS) FROM
IT STARTING AT FIRSTCHAR, THEN REMOVES
ANY SPACES ON THE RIGHT END. RETURNS
INFOGETS AND L (# CHARS. IN INFOGET$).

INFOGETS = MID$(INFO$(INFONO),FIRSTCHAR,NOCHAR)
L = LEN(INFOGET$)
IF L < 1 THEN INFOGET$ = ""; RETURN
IF MID$(INFOGET$,L,1) = " " OR MID$(INFOGET$,L,1) = CHR$(0) THEN L = L - 1:
GOTO 5930
INFOGETS = LEFT$(INFOGET$,L)
RETURN

SUBROUTINE TIMEOUT

IF THE STUDENT TAKES TO LONG TO ANSWER
A 'YESNO' OR A 'CHOICE' QUESTION THIS
SUBROUTINE RE-CALLS STDNSRV.

LOCATE 10,20
PRINT "SORRY....OUT OF TIME.  GOODBYE!!"
LOCATE 11,20:PRINT STRINGS(33,223)
CLOSE
FOR N • 1 TO 3000
NEXT N
RUN

SUBROUTINE WHO'S INTERVIEWING?

THIS SUBROUTINE CREATE A calls
'INTFILE'; WHICH CONTAINS THE FILE NUMBERS
FOR EMPLOYERS WHO WILL BE INTERVIEWING IN
THE NEXT 30 DAYS. IF 'OPENINTV' IS 1, IT
WILL CONTAIN ONLY NUMBERS OF EMPLOYERS FOR
WHOM INTERVIEWS HAVE ALREADY BEEN OPENED.

INDEX • 1
TODAY • VAL(MID$(DATES,9,2))*365 + VAL(MID$(DATES,1,2))*30 + VAL(DATES)
OPEN "R", 1, "B:EMPLCROS", 64
FIELD 1, 64 AS CROSREF
GET 1
IF LOC(1) = MAXEMP + 1 THEN 6390
IF CROSREF$ = STRINGS$(64,0) THEN 6390
IF LEFT$(CROSREF$,62) = STRING$(62,48) THEN 6260
IF LEFT$(CROSREF$,62) = STRING$(62,32) THEN 6260
INDEX = LOC(1)
FIRSTDATE = VAL(MID$(CROSREF$,55,2))*365 + VAL(MID$(CROSREF$,51,2))*30 + VAL(MID$(CROSREF$,53,2))
LASTDATE = VAL(MID$(CROSREF$,61,2))*365 + VAL(MID$(CROSREF$,57,2))*30 + VAL(MID$(CROSREF$,59,2))
IF TODAY + 30 < FIRSTDATE THEN 6260
IF TODAY > LASTDATE THEN 6260
IF OPENINTV = 1 AND MID$(CROSREF$,63,1) <> "1" THEN 6260
INDEX = INDEX + 1
GOTO 6260
CLOSE 1
INDEX = INDEX - 1
6410 RETURN
6420 REM -------------------------------
6430 REM STUDENT'S QUALIFICATIONS
6440 REM -------------------------------
6450 REM THIS SUBROUTINE GETS THE STUDENT'S
6460 REM QUALIFICATIONS FROM INFO(2). IT
6470 REM CHECKS DEGREE, MAJOR, G.P.A.,
6480 REM CITIZENSHIP, AND GRADUATION DATE.
6490 REM
6500 FIRSTCHAR = 9: NOCHAR = 1: INFON0 = 2
6510 GOSUB 5900
6520 DEGREES = INFOGET
6530 FIRSTCHAR = 4: NOCHAR = 4
6540 GOSUB 5900
6550 MAJOR$ = INFOGET
6560 FIRSTCHAR = 10: NOCHAR = 3
6570 GOSUB 5900
6580 GPA = VAL(INFOGET$)
6590 FIRSTCHAR = 8: NOCHAR = 1
6600 GOSUB 5900
6610 CITIZSHIP = VAL(INFOGET$)
6620 FIRSTCHAR = 1: NOCHAR = 3
6625 GOSUB 5900
6630 GRADDATE$ = INFOGET$  
6640 RETURN
6650 REM THESE ARE ERROR TRAPS FOR PRINTER.
6660 REM
6670 REM
6675 REM THESE ARE ERROR TRAPS FOR PRINTER.
6680 REM
6690 REM
6700 PRINT " PLEASE CHECK THE PRINTER";PRINT"
6710 RESUME 4460
6720 PRINT " PLEASE CHECK THE PRINTER";PRINT"
6730 RESUME 5040
6740 PRINT " READY TO CONTINUE? (Y/N) ";GOSUB 5200
6750 Ok
This appendix presents a functional description of EMPLUP and a complete program listing. The use of EMPLUP is described in Chapter 5. For a better understanding of the program, that material should be reviewed before reading on.

**Functional Description**

EMPLUP creates and updates employer files using two "forms" which the user "fills out" as if they were on paper. The information for drawing the forms and getting the data from them is contained in two sequential files, PAGE3 and PAGE4.

The EMPLUP listing begins by defining a number of important variables and dimensioning the three arrays which contain the form drawing information. In line 600, the maximum number of employer files which can be examined is set to 100. This will require a change when greater file space becomes available. If PAGENO is one, the first form is drawn, if two, the second.

The opening menu is given in lines 620-680, and the choice is recorded in the variable CHOICE, which retains
its value through the program execution in order to control program routing.

The first form is displayed and data is put on it in the section of code starting at line 770 for the first choice and at 900 for the second. After the form is completed (line 1100) the second branching point is reached. Subroutine 3500 allows the user to select to edit the data, file it as shown, return to the last menu, or delete the entire file. This choice is recorded as SELECT, which also retains its value in order to determine program routing. The various subroutines will be discussed later. Notice the use of logic to determine routing in lines 1330 to 1350, etc. Also notice that if the option to delete the file is selected, a second opportunity is given to the user to change his or her mind in lines 1510-1530. At line 1290 the page number is incremented in order to complete the second form, and at line 1330 it is determined that both pages are completed, so that eventually at line 1420 the program returns to the main menu.

Subroutines 3000 and 3500 place instructions at the top of the form. In the first case, the instructions tell how to fill out the blank form, and in the second, the optional ways of treating the information displayed are given.
Subroutine 4000 takes the form drawing information from PAGETHRE or PAGEFOUR and loads it into the three arrays CHAR$, LOCAT% and DRW%. The creation of these three arrays is described in Chapter 7. PAGENO controls which file is opened (line 4120) and that the three arrays are filled sequentially with the "stoppers" (asterisk or negative one) telling the program when the array is complete.

Subroutine 4500 actually draws the form on the screen. The larger loop pulls the page drawing information out of DRW% in order: first the row, then the column, then the number of times to print the character, and finally, the character number (in the array CHAR$) to be displayed. The smaller loop from 4720 to 4740 then puts the character on the screen.

After the form has been filled out, subroutine 5000 gathers the data from the form using the locating information given in the array LOCAT%. The first page information is read into INFO$(1) and the second into INFO$(2) as determined by the value of PAGENO. In lines 5220-5320 the subroutine is merely moving the cursor down the form as the user fills the form out. The same logic is used in the second part of the subroutine starting with line 5370 when the data is actually read from the screen.
The data is pulled out of LOCAT% in the order of row, column, number of characters to get, and up to two column numbers to skip over in the process. In line 5500 the first character is located. It and subsequent characters are taken up in TEMPINFO$ so that lines 5560 through 5620 can move any leading blanks to the rear of the string. The subroutine returns after INFO$ is complete for the page number specified.

Subroutine 6000 complements subroutine 5000 in that it takes data out of INFO$ and puts it on the form. This subroutine works very similarly and need not be discussed. Exit from the subroutine occurs in line 6210 when the "stopper" in LOCAT% is found.

Subroutine 7000 will use either the employer name or accept the file number in order to find the employer file. Since this subroutine uses the employer cross file, it is the one used to issue a warning message if the interview files for this employer have already been opened. Getting the number or the employer name uses similar logic to that already discussed. Lines 7320-7410 are used to search for a cross file with the given employer name. Besides removing leading blanks, nothing can be done to insure that the user spells out the company name exactly the way it is shown in the employer cross file. The interchanges in lines 7430-7530 take care of the user's possibly not being able to find the exact name sought.
Subroutine 8000 takes the employer file number, retrieved in the previous subroutine, opens the employer file and puts the data into INFO$(1, 2, and 3).

Subroutine 9000 is called in order to delete an existing file. Before it is called, a confirmation message was given as previously discussed. Since the cross file is always accessed first, putting all zeros in it is enough to indicate that both files are free.

Subroutine 10000 finds an open or never-used employer file by checking for zeros or nulls in the employer cross file. Line 10130 checks to see if the contents of the cross file picked up are all zeros or nulls. The subroutine returns FILENO equal to the file number found. It will stop searching at MAXEMP.

Subroutine 11000 updates both the employer and employer cross file with two exceptions. The employer cross file interviews opened flag is left unchanged—it is set by UPDAY. The employer file page 3 information is not affected—it is stored when UPDAY opens interviews and can only be changed by students signing up for interviews using STDNSRV. The only aspect which needs explanation is the use of the variable "D" in the first part of the subroutine as a pointer to the interview date. Since the last interview date (as well as the first) must go into the
cross file, the possible dates are searched from the last possible date (characters 161 through 166) backwards until the actual last interview date is found. (Line 11240 determines that the date is not all zeros, not all blanks and not all null characters so it must be a real date.) That date is then put into the cross file information by line 11250 and the cross file is closed. The rest of the subroutine is self-explanatory.
EMPLUP Listing

LIST
10 REM EMPLUP.BAS 9-07-84 ORIGINAL JOHN C. DEBO
20 REM COMPLETE REVISION 10-20-84
30 REM DEBUG WORK LATEST 11-3-84
40 REM Tinkering and renumbered 11-20-84
50 REM Corrected file prob. in 7000 1-20-85
60 REM Change EMPLOYER p. 2 is 177 bytes 2-2-85
70 REM
80 REM
90 REM THIS PROGRAM IS CALLED BY EXECUTIV.BAS
100 REM IT IS USED BY THE STAFF TO CREATE AND UPDATE
110 REM EMPLOYER FILES. SOFTWARE FILES NAMED PAGETHREE,
120 REM PAGEFOUR, EMPLCROS AND EMPLYRS ARE ALL USED.
130 REM
140 REM
150 REM ARRAY CHARS HOLDS GRAPHICS CHARACTERS AND WORDING
160 REM USED IN THE CREATION OF THE "FORM" WHICH THE
170 REM USER FILLS OUT.
180 REM
190 REM ARRAY LOCAT% HOLDS THE LOCATIONS OF THE DATA FILLED
200 REM IN BY THE USER FOR PROGRAM RETRIEVAL.
210 REM
220 REM ARRAY DRWX CONTAINS LOCATIONS AND NUMBER OF CHARACTERS
230 REM REQUIRED WHEN THE PROGRAM DRAWS THE FORM ON THE SCREEN.
240 REM
250 REM THE SEQUENTIAL FILES PAGETHRE & PAGEFOUR CONTAIN ALL
255 REM INFORMATION REQUIRED TO DRAW THE PAGES. THEY ARE
256 REM CREATED BY CREAPG3 & CREAPG4.
260 REM
270 REM INFO%(1,2 ETC.) STORES THE DATA BEING TRANSFERRED TO
280 REM AND FROM THE DATA FILE CALLED "EMPLYRS".
290 REM PAGENO DETERMINES WHICH PAGE OF INFO% IS ACCESSED.
300 REM
310 REM ROW AND COL ALWAYS REFER TO THE SCREEN LOCATION;
320 REM ROWS 1 TO 25 AND COLUMNS 1 TO 80.
330 REM
340 REM MAXEIP CONTAINS THE MAXIMUM SIZE OF THE EMPLYRS
350 REM AND EMPLCROS FILES. IT SHOULD BE SET TO REFLECT
360 REM THE STORAGE MEDIUM THESE PROGRAMS ARE RUN ON.
370 REM
380 REM FLAG = 1 INDICATES THE USER WISHED TO RETURN
390 REM TO THE MENU RATHER THAN COMPLETE THE CURRENT
400 REM SUBROUTINE. ANY DATA IN ARRAYS WOULD BE LOST.
410 REM
420 REM CHOICE AND SELECT CONTROL PROGRAM FLOW AMONG THE
430 REM VARIOUS SUBROUTINES. PAGENO ALSO SERVES A FLOW
440 REM CONTROL FUNCTION--IT DETERMINES WHETHER THE
450 REM DATA IN INFO% IS READY TO FILE OR STILL NEEDS
460 REM THE SECOND PAGE.
470 REM
480 REM FILEIT CHOOSES THE SECOND FUNCTION OF SUBROUTINE
490 REM 5000--THAT OF JUST FILING THE DATA FROM SCREEN.
500 REM
510 REM
520 REM SET ARRAY SIZES AND INITIALIZE FLAGS
530 REM
540 REM SET STORAGE MEDIUM LIMITS
540 DIM CHAR$(60)
550 DIM LOCATX(250)
560 DIM DRW$(800)
570 PAGENO = 1
580 FILEIT = 0
590 FLAG = 0
600 MAXEMP = 100
610 REM -----------------------------------
620 CLS
630 PRINT "THIS PROGRAM EDITS OR CREATES NEW EMPLOYER FILES"
640 PRINT
650 PRINT "DO YOU WISH TO:"
660 PRINT " 1) EDIT/DELETE AN EXISTING EMPLOYER FILE?"
670 PRINT " 2) CREATE A NEW EMPLOYER FILE?"
680 PRINT " 3) RETURN TO THE MAIN MENU?"
690 SOUND 523,3
700 PRINT "CHOOSE (1,2,OR 3) PLEASE =>";
710 INPUT "",CHOICE$;
720 CHOICE = FIX(VAL(CHOICE$))
730 IF CHOICE<1 OR CHOICE>3 THEN SOUND 100,4; GOTO 700
740 REM ON CHOICE GOTO 820, 960, 1050
750 REM
760 REM CHOICE # 1 EDIT/DELETE EXISTING FILE
770 REM -------------------------------------
780 REM THIS ROUTINE FINDS THE FILE, PUTS THE ORIGINAL DATA UP (PAGE 3 & 4), AND ALLOWS THE DATA TO BE UPDATED.
790 REM THE PAGE NUMBER
800 REM GOSUB 7000 'GOT FILE NUMBER
810 REM IF FLAG = 1 THEN FLAG = 0; GOTO 570
820 REM GOSUB 8000 'GET FILE DATA
830 REM GOSUB 4000 'GET PAGE DWG DATA
840 REM GOSUB 4500 'DRAW PAGE
850 REM GOSUB 6000 'PUT DATA ON PAGE
860 REM GOTO 1100
870 REM
880 REM CHOICE # 2 CREATE A NEW EMPLOYER FILE
890 REM ----------------------------------------
900 REM THIS ROUTINE GETS PAGE DWG DATA, DRAWS THE PAGE, AND GETS THE NEW EMPLOYER'S DATA.
910 REM
920 REM GOSUB 4000 'GET PAGE DWG DATA
930 REM GOSUB 4500 'DRAW THE PAGE
940 REM GOSUB 3000 'MESSAGE #1
950 REM GOSUB 5000 'GET DATA FROM PAGE
960 REM IF FLAG = 1 THEN FLAG = 0; GOTO 620
970 REM GOTO 1100
980 REM
990 REM CHOICE # 3 RETURN TO MAIN MENU
1000 REM -----------------------------------
1010 REM RUN "A:EXECUTIVE.BAS"
1020 REM
1030 REM THE 1ST TWO OPTIONS COME TOGETHER HERE
1040 REM ------------------------------------------
1050 REM
1060 REM
1070 REM GOSUB 3500 'MESSAGE # 2
1080 REM INPUT "",SELECT$;
1090 REM SELECT = FIX(VAL(SELECT$))
1100 REM IF SELECT<1 OR SELECT>4 THEN SOUND 100,4; GOTO 1100
1110 REM ON SELECT GOTO 1180, 1260, 1460, 1500
1150 REM
1160 REM  SELECT # 1 EDIT THE DATA
1170 REM  ------------------------
1180 ON CHOICE GOTO 1210, 1190
1190 GOSUB 4500 'REDRAW THE PAGE
1200 GOSUB 6000 'PUT THE DATA BACK UP
1210 GOSUB 5000 'TAKE DATA FROM PAGE
1220 GOTO 1100
1230 REM
1240 REM  SELECT # 2 FILE THE DATA
1250 REM  ------------------------
1260 FILEIT = 1
1270 GOSUB 5000 'TAKE DATA FROM PAGE
1280 FILEIT = 0
1290 PAGENO = PAGENO + 1
1300 REM
1310 REM  'WHEN PAGENO = 3 FILE ALL
1320 REM
1330 IF PAGENO >= 3 THEN 1380
1340 IF CHOICE = 2 THEN 960
1350 IF CHOICE = 1 THEN 850
1360 CLS1 PRINT ">>>>>> PROGRAM ERROR <<<<<<"; PRINT "CHOICE ="; CHOICE; END
1370 REM
1380 IF CHOICE = 1 THEN 1410
1390 GOSUB 10000 'GET AN OPEN FILE
1400 IF FLAG = 1 THEN 570
1410 GOSUB 11000 'UPDATE EMPLYRS & EMPLCROS
1420 GOTO 570
1430 REM
1440 REM  SELECT # 3 RETURN TO MENU
1450 REM  ------------------------
1460 GOTO 570
1470 REM
1480 REM  SELECT # 4 DELETE THE FILE
1490 REM  ------------------------
1500 LOCATE 1, 1
1510 PRINT "DELETE THE FILE?"
1520 PRINT "ARE YOU SURE?"
1530 PRINT "TYPE Y OR N AND <RETURN>";
1540 SOUND 523, 3
1550 INPUT "", N$
1560 IF N$ <> "Y" AND N$ <> "y" THEN 1100
1570 IF CHOICE = 2 GOTO 1590
1580 GOSUB 9000 'DELETE THE CURRENT FILE
1590 GOTO 570
1600 REM
1610 REM  THIS IS THE END OF THE MAIN PROGRAM
1620 REM  ------------------------
1630 REM
1640 REM  SUBROUTINE 3000
1650 REM
1660 REM  THIS SUBROUTINE PRINTS MESSAGE #1
1670 REM
1680 REM  USED TO INFORM THE USER ABOUT
1690 REM  ENTERING DATA UNDER OPTION #2
1700 REM  ------------------------
1710 LOCATE 1, 1
PRINT "ENTER INFORMATION WITHIN APPROPRIATE FIELDS USING <RETURN> TO MOVE BETWEEN FIELDS. DO NOT USE COMMAS! YOU ALSO MAY ENTER "SELECT" AND <RETURN> TO DISCARD THE INFORMATION ENTERED AND RETURN TO THE LAST MENU.

SOUND 523,3
RETURN

---------------------------------SUBROUTINE 3500
THIS SUBROUTINE PRINTS MESSAGE #2 AND THE PROMPT. IT'S USED TO GET "SELECT" WHICH CONTROLS THE USE OF THE DATA.

LOCATE 1,1
PRINT "YOU MAY NOW: 1) EDIT THE DATA 2) FILE THE DATA AS SHOWN 3)RETURN TO LAST MENU 4)DELETE THE FILE CHOOSE (1 TO 4) PLEASE ==>
SOUND 523,3
RETURN

---------------------------------SUBROUTINE 4000
THIS ROUTINE LOADS "PAGE" INFORMATION INTO THE APPROPRIATE ARRAYS FOR CREATION OF THE DATA ENTRY FORMS (PAGETHRE AND PAGEFOUR) P1 ALWAYS POINTS TO THE NEXT ARRAY LOCATION TO BE FILLED. THE THREE ARRAYS ARE DESCRIBED AT THE START OF THIS PROGRAM.

ON PAGENO GOTO 4130,4150
OPEN "I",1,"B:PAGETHRE"
GOTO 4160
OPEN "I",1,"B:PAGEFOUR"
Pl+ 1
INPUT 11, CHAR$(P1>
IF CHAR$(P1) • "*" THEN Pl+ 1
Pl+ 1
INPUT LOCATY(P1>
IF LOCATY(P1) • -1 THEN Pl+ 1
Pl+ 1
INPUT DRWY(P1>
IF DRWY(P1) • -1 THEN Pl+ 1
4170 INPUT #1, CHAR$(P1)
4180 IF CHAR$(P1) = "#" THEN 4210
4190 P1 = P1 + 1
GOTO 4170
4210 P1 = 1
4220 INPUT #1, LOCATX(P1)
4230 IF LOCATX(P1) = -1 THEN 4260
4240 P1 = P1 + 1
GOTO 4220
4260 P1 = 1
4270 INPUT #1, DRWX(P1)
4280 IF DRWX(P1) = -1 THEN 4310
4290 P1 = P1 + 1
GOTO 4270
4310 CLOSE 1
4320 RETURN

4500 REM SUBROUTINE 4500
4510 REM THIS ROUTINE DRAWS THE FORM TO BE FILLED
4520 REM OUT WITH THE STUDENT INFORMATION.
4530 REM DRW% CONTAINS THE ROW, THE COLUMN (COL),
4540 REM THE NUMBER OF REpetitions OF A CHARACTER
4550 REM (NPRNT), AND THE CHARACTER NUMBER (NCHAR).
4560 REM
4570 REM
4580 REM
4590 REM
4600 CLS
4610 P1 = 1
4620 ROW = DRW%(P1)
4630 P1 = P1 + 1
4640 COL = DRW%(P1)
4650 P1 = P1 + 1
4660 NPRNT = DRW%(P1)
4670 P1 = P1 + 1
4680 NCHAR = DRW%(P1)
4690 IF NCHAR = -1 THEN RETURN
4700 P1 = P1 + 1
4710 LOCATE ROW, COL
4720 FOR P2 = 1 TO NPRNT
4730 PRINT CHAR$(NCHAR);
4740 NEXT P2
4750 GOTO 4620

5000 REM SUBROUTINE 5000
5010 REM THIS SUBROUTINE BATHERS THE DATA ENTERED
5020 REM ON THE FORM.
5030 REM LOCAT% CONTAINS ROW, COLUMN (COL), THE
5040 REM NUMBER OF CHARACTERS TO BE READ (NCHARS),
5050 REM AND ANY COLUMNS TO SKIP WHILE PICKING UP
5060 REM THE DATA (NSKIPI & NSKIP2).
5070 REM
5080 REM INFO$(1 AND 2) HOLD THE INFORMATION READ IN.
5090 REM
5100 REM P1 POINTS TO THE LOCATION IN INFO$ BEING
5110 REM USED. P2 POINTS TO THE CHARACTER NUMBER.
5120 REM
5130 P1 = 1
5140 P2 = 1
5150 INF0$(PAGENO) = "" 'EMPTY OUT INFO$
5160 IF FILEIT = 1 THEN 5370
5170 FIRST TIME THROUGH LOCAT%—JUST MOVING CURSOR
5180 INF0$(PAGENO) = "" 'EMPTY OUT INFO$
5190 REM
5200 ROW = LOCAT%(P1)
5210 P1 = P1 + 1
5220 COL = LOCAT%(P1)
5230 P1 = P1 + 3
5240 NSKIP2 = LOCAT%(P1)
5250 P1 = P1 + 1
5260 IF NSKIP2 = -1 THEN 5370
5270 LOCATE ROW, COL
5280 INPUT "", B$
5290 IF B$ = "MENU" OR B$ = "menu" THEN FLAG = 1: RETURN
5300 GOTO 5210
5330 REM THE SECOND TIME THROUGH LOCATX PICKS UP
5340 REM INFORMATION FROM THE SCREEN FOR INFO$.
5350 REM AND REMOVES LEADING BLANKS IN EACH FIELD.
5360 REM
5370 P1 = 1
5380 ROW = LOCATX(P1)
5390 TEMPINFO$ = "" 'EMPTY TEMPORARY INFO$
5400 P1 = P1 + 1
5410 COL = LOCATX(P1)
5420 P1 = P1 + 1
5430 NCHARS = LOCATX(P1)
5440 P1 = P1 + 1
5450 NSKIP1 = LOCATX(P1)
5460 P1 = P1 + 1
5470 NSKIP2 = LOCATX(P1)
5480 IF NSKIP2 = -1 THEN RETURN
5490 P1 = P1 + 1
5500 LOCATE ROW,COL 'FIND THE FIELD
5510 TEMPINFO$ = TEMPINFO$ + CHR$(SCREEN(ROW,COL))
5520 COL = COL + 1 'LOAD FIELD INTO TEMPINFO$
5530 IF COL = NSKIP1 OR COL = NSKIP2 THEN COL = COL + 1
5540 P2 = P2 + 1
5550 IF P2 <= NCHARS THEN 5510
5550 X = ASC(TEMPINFO$) 'WHAT'S FIRST CHAR?
5550 IF X <> 32 THEN 5630 'NO LEADING BLANKS
5560 NCHARS = NCHARS - 1 'LOOP COUNTER
5570 TEMPINFO$ = M$ + " " 'REMOVE LEADING BLANK
5580 TEMPINFO$ = M$ + " " 'AND PUT IT ON RIGHT
5590 IF NCHARS <= 1 THEN 5630
5600 IF NCHARS <> 1 THEN 5630
5610 IF NCHARS = 1 THEN 5630
5620 GOTO 5560
5630 INFO$(PAGENO) = INFO$(PAGENO) + TEMPINFO$
5640 P2 = P2 + 1
5650 GOTO 5380

6000 REM SUBROUTINE 6000
6010 REM THIS SUBROUTINE PUTS DATA FROM INFO$(1 OR 2)
6020 REM ONTO THE EMPLOYER FORM.
6030 REM VARIABLES USED ARE THE SAME AS THE PREVIOUS
6040 REM SUBROUTINE
6050 REM
6060 REM
6100 P1 = 1
6110 P2 = 1
6120 ROW = LOCATX(P1)
6130 P1 = P1 + 1
6140 COL = LOCATX(P1)
6150 P1 = P1 + 1
6160 NCHARS = LOCATX(P1)
6170 P1 = P1 + 1
6180 NSKIP1 = LOCATX(P1)
6190 P1 = P1 + 1
6200 NSKIP2 = LOCATX(P1)
6210 IF NSKIP2 = -1 THEN RETURN
6220 P1 = P1 + 1
6230 LOCATE ROW,COL
6240 PRINT MID$(INFO$(PAGENO),P2,1);
6250 P2 = P2 + 1
6260 NCHARS = NCHARS - 1
6270 IF NCHARS < 1 THEN 6120
6280 COL = COL + 1
6290 IF COL = NSKIP1 OR COL = NSKIP2 THEN PRINT "-"; COL = COL + 1
6300 GOTO 6240
7000 REM SUBROUTINE 7000
7010 REM THIS SUBROUTINE FINDS THE PROPER
7020 REM EMPLOYER FILE USING EITHER THE FILE
7030 REM NUMBER OR THE EMPLOYER NAME.
7040 REM
7050 REM
7060 CLS
7070CLS
7080 PRINT "DO YOU KNOW THE EMPLOYER'S FILE NUMBER?"
7090 PRINT "ENTER Y OR N ===>";
7100 SOUND 523,3
7110 INPUT "", N$
7120 IF N$ = "y" OR N$ = "Y" THEN 7140
7130 IF N$ <> "N" AND N$ <> "n" THEN 7070 ELSE 7180
7140 PRINT "ENTER THE FILE NO. ===>";
7150 SOUND 523,3
7160 INPUT "", FILENO
7170 GOTO 7550
7180 PRINT "ENTER THE EMPLOYER'S NAME EXACTLY AS IT APPEARS IN THE FILES"
7190 PRINT "EMPLOYER'S NAME ===>";
7200 SOUND 523,3
7210 INPUT "", N$
7220 N$ = N$ + SPACE$(25)
7230 NAM$ = LEFT$(N$, 25) 'GIVES A 25 CHAR STRING
7240 NCHARS = 25
7250 X = ASC(NAM$) 'GET THE FIRST CHARACTER
7260 IF X <> 32 THEN 7320
7270 NCHARS = NCHARS - 1
7280 NAM$ = MID$(NAM$, 2) 'IF IT'S A SPACE, MOVE
7290 NAM$ = N$ + " " 'IT TO THE RIGHT END
7300 IF NCHARS = 0 THEN 7320
7310 GOTO 7250
7320 OPEN "R", 1, "B:EMPLCROS", 64
7330 FIELD 1, 64 AS CROSREF$
7340 GET 1
7350 IF LOC(1) = MAXEMP + 1 THEN 7430
7360 IF CROSREF$ = STRING$(64, 0) THEN 7430
7370 IF LEFT$(CROSREF$, 25) <> NAM$ THEN 7340
7380 REM NUL CHARACTERS INDICATES VIRGIN TERY
7390 REM NO NEED TO CONTINUE LOOKING.
7400 FILENO = LOC(1)
7410 CLOSE 1
7420 GOTO 7550
7430 PRINT "THERE IS NO EMPLOYER NAME IN THE FILE EXACTLY AS GIVEN."
7440 PRINT "--PERHAPS IT IS SPELLED DIFFERENT, ABBREVIATED, OR"
7450 PRINT "HAS A DIFFERENT NUMBER OF SPACES WITHIN IT?"
7460 PRINT
7470 PRINT
7480 SOUND 100, 4
7490 CLOSE 1
7500 PRINT "DO YOU WISH TO RETURN TO THE PREVIOUS MENU? (Y OR N) ===>";
7510 INPUT "", N$
7520 IF N$ <> "Y" AND N$ <> "y" THEN 7080
7530 FLAG = 1
7540 RETURN
7550 REM AT THIS POINT WE HAVE A VALID FILENO FROM
7560 REM EITHER THE USER OR EMPLCROS. NOW CHECK THE
7570 REM DATE INTERVIEWS OPEN IN CASE A WARNING NEEDED.
7580 OPEN "R",1, "B:EMPLCROS",64
7590 FIELD 1, 64 AS CROSREF$  
7600 BET 1,FILENO
7610 IF MID$(CROSREF$,63,1) <> "1" THEN CLOSE 1:RETURN
7620 PRINT "YOU'RE EDITING OR DELETING A FILE WHICH HAS ALREADY BEEN OPENED"
7630 PRINT "FOR STUDENT INTERVIEWS. YOU MUST BE SURE THAT NO STUDENT"
7640 PRINT "WILL BE AFFECTED BY YOUR CHANGES."
7645 CLOSE 1
7650 RETURN
8000 REM -----------------------------------SUBROUTINE 8000
8010 REM THIS SUBROUTINE LOADS DATA FROM EMLPYS
8020 REM INTO INFOS(1,2 AND 3).
8030 REM
8040 OPEN "R",2, "B:EMPLPYS", 512
8050 FIELD 2, 205 AS P61INFO$, 177 AS P62INFO$, 130 AS PG3INFO$
8060 BET 2, FILENO
8070 INFO$(1) = P61INFO$
8080 INFO$(2) = P62INFO$
8090 INFO$(3) = PG3INFO$
8100 CLOSE 2
8110 RETURN
9000 REM -----------------------------------SUBROUTINE 9000
9010 REM THIS SUBROUTINE IS FOR DELETING AN
9020 REM EXISTING FILE. DOUBLE CHECK WITH
9030 REM USER BEFORE YOU CALL IT!!
9040 RETURN
9050 REM
9060 OPEN "R",1, "B:EMPLCROS",64
9070 FIELD 1, 64 AS CROSREF$
9080 N$ = STRINGS$(64,48)
9090 LSET CROSREF$ = N$
9100 PUT 1, FILENO
9110 CLOSE 1
9120 RETURN
10000 REM -----------------------------------SUBROUTINE 10000
10010 REM THIS SUBROUTINE FINDS AN OPEN FILE
10020 REM A NEVER USED FILE CONTAINS NULLS
10030 REM AN ERASED FILE CONTAINS ZEROS
10040 REM ONLY EMPLCROS IS TESTED.
10050 REM
10060 OPEN "R",1, "B:EMPLCROS",64
10070 FIELD 1, 64 AS CROSREF$
10080 GET 1
10090 IF CROSREF$ <> STRINGS$(64,0) AND CROSREF$ <> STRINGS$(64,48) THEN 10120
10100 IF FILENO = LOC(1)
10110 IF FILENO <= MAXEMP THEN 10270
10120 PRINT
10130 PRINT "EMPLOYER FILE SPACE IS FULL--CAN NOT FILE"
10200 PRINT "DELETE SOME EMPLOYER FILES AND RE-ENTER THIS NEW FILE."
10210 SOUND 100,4
10220 CLOSE 1
10230 GOSUB 12000 'STRIKE ANY KEY
10240 FLAG = 1
10250 RETURN
10260 PRINT
10270 PRINT
10280 PRINT "THE EMPLOYER WILL BE GIVEN FILE NUMBER ";FILENO;"--YOU WILL "
10290 PRINT "WANT TO RETAIN THE NUMBER IN YOUR RECORDS."
10300 PRINT "STRIKE ANY KEY
10310 PRINT
10320 CLOSE 1
10330 GOSUB 12000 'STRIKE ANY KEY
10340 RETURN
11000 REM ------------------------------SUBROUTINE 11000
11010 REM
11020 REM THIS SUBROUTINE UPDATES THE EMPLYRS
11030 REM AND EMPLCROS FILES. ALL INFORMATION
11040 REM IN BOTH FILES IS REWRITTEN EXCEPT--
11050 REM PB3INFO (INTERVIEW SCHEDULE INFO) AND
11060 REM THE FLAG IN STONCROS INDICATING THAT
11070 REM THE INTERVIEW FILES HAVE BEEN OPENED.
11080 REM
11090 REM
11100 REM
11110 OPEN "R", 1, "B;EMPLCROS",64
11120 FIELD 1, 64 AS CROSREF*
11130 GET 1,FILENO
11140 M$ = MID$(CROSREF*,63,1) 'GET INTERVIEWS EXIST FLAG
11150 REM
11160 REM FIRST, FIND THE LAST INTERVIEW DATE
11170 REM D POINTS TO THE DATE IN INFO$(2)
11180 REM N$ HOLDS THE DATE FIELD WHILE IT
11190 REM IS CHECKED TO SEE IF ANYTHING IS THERE.
11200 REM
11210 D = 161
11220 N$ = MID$(INFO$(2),D,6) 'DATE FIELD FOR INTERVIEWS
11230 IF D <= 125 THEN 11250
11240 IF N$ = STRING$(6,0) OR N$ = STRING$(6,32) OR N$ = STRING$(6,48) THEN D =
11250 GO TO 11220
11260 LSET CROSREF$ = MID$(INFO$(1),1,50) + MID$(INFO$(2),125,6) +N$ +M$ +""
11270 PUT 1,FILENO
11280 CLOSE 1
11290 REM
11300 REM NOW THE ACTUAL DATA FROM EMPLYRS IS
11310 REM UPDATED. NOTE THAT INFO$(3) IS RETAINED
11320 REM UNCHANGED FROM THE FILE.
11330 REM
11340 OPEN "R", 2, "B;EMPLYRS", 512
11350 FIELD 2, 205 AS PB1INFO$, 177 AS PB2INFO$, 130 AS PB3INFO$
11360 GET 2,FILENO
11370 LSET PB1INFO$ = INFO$(1)
11380 LSET PB2INFO$ = INFO$(2)
11390 PUT 2, FILENO
11400 CLOSE 2
11410 REM
11420 REM SUBROUTINE 12000
11430 REM
12030 REM THIS SUBROUTINE IS CALLED "STRIKE
12040 REM ANY KEY". IT ALLOWS THE USER TIME
12050 REM TO THINK, COPY DOWN INFORMATION, ETC.
12060 REM
12070 PRINT PRINT PRINT
12080 PRINT " >>> PRESS ANY KEY TO CONTINUE <<<"
12090 N$ = ""
12100 WHILE LEN(N$) < 1
12110 N$ = INKEY$
12120 WEND
12130 CLS
12140 RETURN
Ok
APPENDIX C

STDNUP--STUDENT FILE UPDATING PROGRAM

This appendix presents a functional description of STDNUP and a complete program listing. The use of STDNUP is described in Chapter 6.

Functional Description

Much of the inner workings of STDNUP are very similar to the program EMPLUP described in the last appendix; the description here will therefore be brief. As with EMPLUP, this program makes use of two forms to create or update information for the student file. The information for the two forms is contained in the sequential files PAGEONE and PAGETWO.

After describing the important variables and dimensioning the same arrays used in EMPLUP, the program sets the maximum number of student files in line 470. This will need to be changed when a larger storage medium becomes available. Option #2 (line 730 and on), which creates a new file, is clearly separated from the first option, that of updating or deleting an existing file (line 1450 and on).

The procedure in creating a new file is identical to that used in EMPLUP. First, the page drawing information is
gotten using subroutine 4000, then subroutine "Page handle" is called. This subroutine calls subroutine 4500 to actually draw the form, subroutine 5000 to get the data from the form, and then ascertains what the user wishes done with it. Depending upon the choice in line 3300, the data is filed, discarded, or put back on the screen again for further editing. When data is to be put onto the screen, subroutine 6000 does it.

   The function of the subroutines is nearly identical to those in the previous appendix, and their description should be referred to for further information.

   Under option #1 in lines 1630-1680, the student's social security number is gotten, and the student cross files are searched (lines 1690-1760) for the correct file. Aside from calling subroutine 6000 after the page has been drawn in order to put up the information already held in the file, there are no major differences between the operation of the two options.
LIST
10 REM STDNUP.BAS 8-11-84 ORIGINAL JOHN C. DEBO
20 REM REVISED 10-19-84 ADD DATING OF STDNCROS
30 REM REVISED 10-21-84 ADD FILENO QUESTION
40 REM REVISED 11-2-84 Line 5518 MID
50 REM Tinkering and renumbered 11-20-84
60 REM FIX PROBLEM IN STDNCROS UPDATE 1-20-85
70 REM
80 REM
90 REM -------------------------------------------------------------
100 REM THIS PROGRAM IS CALLED BY EXECUTIV.BAS
110 REM IT IS USED BY THE STAFF TO CREATE AND UPDATE
120 REM STUDENT FILES. SOFTWARE FILES NAMED PAGEONE,
130 REM PAGETWO, STDNCROS, AND STUDENTS ARE ALL USED.
140 REM -------------------------------------------------------------
150 REM
160 REM ARRAY CHAR* HOLDS GRAPHICS CHARACTERS AND WORDING
170 REM USED IN THE CREATION OF THE "FORM" WHICH THE
180 REM USER FILLS OUT.
190 REM
200 REM ARRAY LOCAT% HOLDS THE LOCATIONS OF THE DATA FILLED
210 REM IN BY THE USER FOR PROGRAM RETRIEVAL.
220 REM
230 REM ARRAY DRW% CONTAINS LOCATIONS AND NUMBER OF CHARACTERS
240 REM REQUIRED WHEN THE PROGRAM DRAWS THE FORM ON THE SCREEN.
250 REM
260 REM THE FORMS (PAGEONE & PAGETWO) ARE SEQUENTIAL FILES
261 REM CREATED BY CREAPG1 & CREAPG2. THEY CONTAIN THE INFORMATION
262 REM NEEDED TO DRAW THE PAGES USED IN INPUT/OUTPUT OF DATA.
270 REM
280 REM INFO% (1 OR 2) STORES THE DATA BEING TRANSFERRED TO
290 REM AND FROM THE DATA FILE CALLED "STUDENTS".
300 REM PAGENO DETERMINES WHICH PAGE OF INFO% IS ACCESSED.
310 REM
320 REM ROW AND COL ALWAYS REFER TO THE SCREEN LOCATION;
330 REM ROWS 1 TO 25 AND COLUMNS 1 TO 80.
340 REM
350 REM MAXSTU CONTAINS THE MAXIMUM SIZE OF THE STUDENTS
360 REM AND STDNCROS FILES. IT SHOULD BE SET TO REFLECT
370 REM THE STORAGE MEDIUM THESE PROGRAMS ARE RUN ON.
380 REM
390 REM FLAG = 1 INDICATES THE USER WISHED TO RETURN
400 REM TO THE MENU RATHER THAN COMPLETE THE CURRENT
410 REM SUBROUTINE. ANY DATA IN ARRAYS WOULD BE LOST.
420 REM
430 REM -------------------------------------------------------------
440 DIM CHAR*(60)
450 DIM LOCAT%(150)
460 DIM DRW%(450)
470 MAXSTU = 100
480 REM
490CLS
500 FLAG = 0
510 PRINT "THIS SECTION OF THE PROGRAM EDITS OR CREATES"
520 PRINT "NEW STUDENT FILES."
530 PRINT
540 PRINT "DO YOU WISH TO:"
550 PRINT "1) EDIT/DELETE AN EXISTING STUDENT FILE?"
560 PRINT "2) CREATE A NEW STUDENT FILE?"
570 PRINT "3) RETURN TO THE MAIN MENU?"
580 SOUND 523,3
590 PRINT " CHOOSE (1,2,OR 3) PLEASE ===>");
600 INPUT"",N$
610 N = FIX (VAL (N$))
620 IF N<1 OR N>3 THEN SOUND 100,4: GOTO 590
630 REM
640 ON N GOTO 1450, 800, 690 "###JUMP ADDRESSES###
650 REM
660 REM OPTION # 3 RETURN TO MAIN MENU
670 REM
680 REM
690 RUN "A1EXECUTIV.BAS"
700 END
710 REM
720 REM
730 REM OPTION # 2 CREATE A NEW STUDENT FILE
740 REM
750 REM THIS ROUTINE GETS PAGE FILE THEN CALLS
760 REM 3500 TO PUT PAGE GRAPHICS IN ARRAY FORM
770 REM AND 3000 TO DRAW THE PAGE AND PRINT THE
780 REM SUB-MENUS. (SEE CONTINUED REMARKS BELOW)
790 REM
800 PAGENO = 1
810 OPEN "I", 1, "A:PAGEONE"
820 GOSUB 4000
830 REM
840 REM ((SUBROUTINE TO GET "PAGE" DRAWING DATA))
850 REM
860 CLOSE 1
870 GOSUB 3020
880 REM
890 REM ((SUBROUTINE HANDLE ALL PAGE DATA WORK))
900 REM
910 IF FLAG = 1 THEN 490
920 PAGENO = 2
930 OPEN "I", 1, "A:PEGETWO"
940 GOSUB 4000
950 REM
960 REM ((SUBROUTINE TO GET "PAGE" DRAWING DATA))
970 REM
980 CLOSE 1
990 GOSUB 3020
1000 REM
1010 REM ((SUBROUTINE HANDLE ALL PAGE DATA WORK))
1020 REM
1030 IF FLAG = 1 THEN 490
1040 CLS:LOCATE 10,20
1050 PRINT"FILING THE STUDENT DATA"
1060 LOCATE 12,20: PRINT"====== == ===== == ==";
1070 REM
1080 REM OPEN "STDNCROS" AND FIND AN EMPTY FILE, THEN
1090 REM PUT THE SOC. SEC. NO. AND CURRENT DATE IN IT.
1100 REM
1110 REM STUDENT INFORMATION WILL GO IN "STUDENTS".
1120 REM (AN OPEN "STDNCROS" CONTAINS NULLS OR
1130 REM ZEROS)
1140 REM
1150 USEDATES = LEFT$(DATE$(2) + MID$(DATE$(4,2) + MID$(DATE$(9)
1160 SSN$ = MID$(INFO$(1),31,9)
1170 OPEN "R", 1, "B:STDNCROS", 16
1180 FIELD 1, 16 AS CROSREF$
1190 GET 1
1200 IF CROSREF$ <> STRING$(16,0) AND CROSREF$ <> STRING$(16,48) THEN 1190
1210 FILENO = LOC(1)
1220 IF FILENO < 1 + MAXSTU THEN 1280
1230 PRINT "STUDENT FILE SPACE IS FULL----CAN NOT FILE"
1240 PRINT; PRINT "------------
1250 SOUND 100,4
1260 CLOSE 1
1270 GOTO 530
1280 PRINT "STUDENT NUMBER "; SSN$; " WAS GIVEN FILE NUMBER "; FILENO
1300 PRINT
1310 LSET CROSREF$ = SSN$ + USEDATE$
1320 PUT 1, FILENO
1330 CLOSE 1
1340 REM
1350 REM NOW INFOS( 1 AND 2 ) WILL BE PUT INTO THE
1360 REM STUDENT FILE AT FILENO
1370 REM
1380 OPEN "R", 2, "B:STUDENTS", 256
1390 FIELD 2, 154 AS PG1INFO$, 102 AS PG2INFO$
1400 LSET PG1INFO$ = INFOS( 1 )
1410 LSET PG2INFO$ = INFOS( 2 )
1420 PUT 2,FILENO
1430 CLOSE 2
1440 GOTO 530
1450 REM
1460 REM OPTION # 1 EDIT/DELETE EXISTING FILE
1470 REM
1480 REM THIS ROUTINE GETS THE STUDENT'S SOCIAL SECURITY
1490 REM NUMBER, IT OPENS THE STUDENT FILE AND PUTS
1500 REM THE EXISTING INFORMATION ON THE "PAGE" FORM,
1510 REM AND THEN ALLOWS UPDATING. SEE FURTHER REMARKS
1520 REM WITHIN THE BODY OF THE PROGRAM.
1530 REM
1540CLS
1550 PRINT "DO YOU KNOW THE STUDENT'S FILE NUMBER? (Y OR N) ===>";
1560 SOUND 523,3
1570 INPUT ",N$
1580 IF NO <> "Y" AND N$ <> "y" THEN 1630
1590 PRINT "ENTER THE FILE NUMBER ===>";
1600 SOUND 523,3
1610 INPUT ",FILENO
1620 GOTO 1870
1630 PRINT "ENTER THE STUDENT'S SOCIAL SECURITY NUMBER"
1640 PRINT "IN THIS FORMAT: XXX-XX-XXXX"
1650 PRINT "SOC. SEC. NO. ===>";
1660 SOUND 523,3
1670 INPUT ",N$
1680 SSNCODES=LEFT$(N$,3)+MID$(N$,5,2)+MID$(N$,8)
1690 OPEN "R", 1, "B:STDNCROS", 16
1700 FIELD 1, 16 AS CROSREF$
1710 GET 1
1720 IF LOCA(1) = MAXSTU+1 THEN 1780
1730 IF CROSREF$ = STRING$(16,0) THEN 1780
1740 IF LEFT$(CROSREF$,9) <> SSNCODE$ THEN 1710
1750 FILENO = LOC(1)
1760 CLOSE 1
1770 GOTO 1820
1780 PRINT "THE STUDENT SOC. SEC. NO. GIVEN IS NOT ON FILE"
1790 SOUND 100,4
1800 CLOSE 1
1810 GOTO 530
1820 REM
1830 REM NOTE THAT CROSREF* WILL CONTAIN ZEROS IF A FILE
1840 REM WAS THERE PREVIOUSLY AND WAS ERASED. IT WILL
1850 REM CONTAIN NULL CHARACTERS IF IT IS AN UNUSED FILE.
1860 REM
1870 OPEN "R", 2, "B:STUDENTS", 256
1880 FIELD 2, 154 AS PG1INFO*, 102 AS PG2INFO*
1890 GET 2, FILENO
1900 INFO$(1) = PG1INFO*
1910 INFO$(2) = PG2INFO*
1920 CLOSE 2
1930 PAGEENO = 1
1940 OPEN "I", 1, "A:PAGEONE"
1950 GOSUB 4000
1960 REM
1970 REM ((SUBROUTINE TO GET "PAGE" DRAWING INFO))
1980 REM
1990 REM 1990 CLOSE 1
2000 CLS
2010 GOSUB 4500
2020 REM ((SUBROUTINE FOR DRAWING THE PAGE FORM))
2030 REM
2040 REM
2050 GOSUB 6000
2060 REM
2070 REM ((SUBROUTINE TO PUT DATA ON THE FORM))
2080 LOCATE 1,1
2090 PRINT "YOU MAY NOW: 1) EDIT THE DATA SHOWN"
2100 PRINT "2) DISPLAY THE DATA ON PAGE TWO"
2110 PRINT "3) DELETE THE STUDENT FILE"
2120 SOUND 523,3
2130 LOCATE 4,1: PRINT"
2140 LOCATE 4,1: PRINT " ENTER (1 TO 3) PLEASE===>";
2150 INPUT",N
2160 N = FIX(VAL(N))
2170 IF N < 1 OR N > 3 THEN SOUND 100,4: GOTO 2130
2180 ON N GOTO 2190,2470,2290
2190 GOSUB 5000
2200 REM
2210 REM ((SUBROUTINE TO GATHER DATA FROM FORM))
2220 REM
2230 IF FLAG = 1 THEN 490
2240 LOCATE 4,1: PRINT"
2250 GOTO 2080
2260 REM
2270 REM THIS CODE IS TO DELETE THE FILE CALLED UP
2280 REM
2290 LOCATE 1,1
2300 PRINT " 
" 
2310 PRINT " 
" 
2320 LOCATE 1,1  
2330 PRINT "DELETE THE FILE SHOWN? ARE YOU SURE? (Y OR N)??";  
2340 SOUND 523,3  
2350 INPUT "",N$  
2360 IF N$ <> "Y" AND N$ <> "y" THEN 2080  
2370 OPEN "R", 1, "B:STDNCROS",16  
2380 FIELD 1, 16 AS CROSREF$  
2390 N$ = STRING$(16,48)  
2400 LSET CROSREF$ = N$  
2410 PUT 1, FILENO  
2420 CLOSE 1  
2430 CLS: GOTO 530  
2440 REM  
2450 REM THIS SECTION DISPLAYS AND EDITS PAGE TWO  
2460 REM  
2470 CLS  
2480 PAGENO = 2  
2490 OPEN "I", 1, "A:PAGE2"  
2500 GOSUB 4000  
2510 REM  
2520 REM ((SUBROUTINE TO GET "PAGE" DRAWING DATA))  
2530 REM  
2540 CLOSE 1  
2550 GOSUB 4500  
2560 REM  
2570 REM ((SUBROUTINE TO DRAW THE PAGE))  
2580 REM  
2590 GOSUB 6000  
2600 REM  
2610 REM ((SUBROUTINE TO ENTER THE INFO. ON PAGE))  
2620 REM  
2630 LOCATE 1,1  
2640 PRINT "YOU NOW MAY: 1) EDIT THE DATA SHOWN  
"  
2650 PRINT " 2) ACCEPT THE DATA AND CLOSE THE FILE  
"  
2660 PRINT " 3) RETURN TO THE MENU WITH FILE UNCHANGED  
"  
2670 SOUND 523,3  
2680 LOCATE 4,1: PRINT " 
2690 LOCATE 4,1: PRINT " SELECT (1 TO 3) PLEASE ??";  
2700 INPUT "",N$  
2710 N = FIX(VAL(N$))  
2720 IF N < 1 OR N > 3 THEN SOUND 100,4:GOTO 2680  
2730 ON N GOTO 2740, 2840, 2990  
2740 GOSUB 5000  
2750 REM  
2760 REM ((SUBROUTINE TO GET DATA FROM FORM))  
2770 REM  
LIST 2600-2800  
2780 IF FLAG = 1 THEN 490  
2790 LOCATE 4,1: PRINT"  
"  
2800 GOTO 2630  
2810 REM
2820 REM THIS SECTION UPDATES THE FILE AND CLOSES IT
2830 REM
2840 PRINT "FILING THE STUDENT DATA"
2850 LOCATE 12,20: PRINT " == == == == == == == == == ==
2860 PRINT:PRINT " STUDENT NUMBER " + SSN$ + "; IS STORED AT FILE NUMBER " + FILENO
2870 OPEN "R", 1, "B:STDCROS",16
2880 OPEN "R", 2, "B:STUDENTS", 256
2890 OPEN "R", 1, "B:STDCROS",16 AS CROSREF$ +
2900 PUT 1, FILENO
2910 CLS
2920 GOTO 530
2930 GOSUB 4500
2940 GOSUB 5000
2950 IF FLAG = 1 THEN RETURN
2960 IF N < 1 OR N > 3 THEN SOUND 100,4: GOTO 3290
3340 ON N GOTO 3350,3550,3090
3350 CLS
3360 GOSUB 4500
3370 REM
3380 REM "SUBROUTINE TO DRAW PAGE FORM")
3390 REM
3400 GOSUB 6000
3410 REM
3420 REM "SUBROUTINE TO ENTER INFO( 1 / 2 ) ON PAGE")
3430 REM
3440 LOCATE 1,1
3450 PRINT"YOU MAY NOW: 1) FILE THE DATA AS IT IS SHOWN
3460 PRINT" 2) DISCARD THE FORM AND START OVER
3470 PRINT" 3) RETURN TO THE MENU"
3480 SOUND 523,3
3490 LOCATE 4,1: PRINT"
3500 LOCATE 4,1: PRINT" CHOOSE (1 TO 3) PLEASE ==>");
3510 INPUT ",N"
3520 N = FIX(VAL(N))
3530 IF N < 1 OR N > 3 THEN SOUND 100,4: GOTO 3490
3540 ON N GOTO 3550,3090,3560
3550 RETURN
3560 FLAG = 1
3570 RETURN
4000 REM----------------------------------------
4010 REM----------------------------------------
4020 REM----------------------------------------
4030 REM THIS ROUTINE LOADS "PAGE" INFORMATION INTO THE
4040 REM APPROPRIATE ARRAYS FOR CREATION OF THE DATA
4050 REM ENTRY FORMS (PAGEONE, ETC.).
4060 REM----------------------------------------
4070 REM----------------------------------------
4080 REM P1 ALWAYS POINTS TO THE NEXT ARRAY LOCATION
4090 REM TO BE FILLED. THE THREE ARRAYS ARE DESCRIBED
4100 REM AT THE START OF THIS PROGRAM.
4110 REM
4120 P1 = 1
4130 INPUT #1, CHAR*(P1)
4140 IF CHAR*(P1) = "*" THEN 4170
4150 P1 = P1 + 1
4160 GOTO 4130
4170 P1 = 1
4180 INPUT #1, LOCATX(P1)
4190 IF LOCATX(P1) = -1 THEN 4220
4200 P1 = P1 + 1
4210 GOTO 4180
4220 P1 = 1
4230 INPUT #1, DRWY(P1)
4240 IF DRWY(P1) = -1 THEN 4270
4250 P1 = P1 + 1
4260 GOTO 4230
4270 RETURN
4500 REM----------------------------------------
4510 REM----------------------------------------
4520 REM----------------------------------------
4530 REM THIS ROUTINE DRAW THE FORM TO BE FILLED
4540 REM OUT WITH THE STUDENT INFORMATION.
IT DOES NOT CLEAR THE SCREEN

DRWX CONTAINS THE ROW, THE COLUMN (COL),
THE NUMBER OF REPETITIONS OF A CHARACTER
(NPRNT), AND THE CHARACTER NUMBER (NCHAR).

ROW contains the row, the column (COL), the number of repetitions of a character (NPRNT), and the character number (NCHAR).

LOCATE ROW, COL
FOR P2 = 1 TO NPRNT
PRINT CHAR(NCHAR);
NEXT P2
GOTO 4630

-------------------------------------
SUBROUTINE 5000
-------------------------------------
THIS SUBROUTINE GATHERS THE DATA ENTERED ON THE STUDENT FORM.
LOCATX CONTAINS ROW, COLUMN (COL), THE NUMBER OF CHARACTERS TO BE READ (NCHARS), AND ANY COLUMNS TO SKIP WHILE PICKING UP THE DATA (NSKIP1 & NSKIP2).
INFO(1 AND 2) HOLD THE INFORMATION READ IN.

LOCATE ROW, COL
FOR P2 = 1 TO NPRNT
PRINT CHAR(NCHAR);
NEXT P2
GOTO 4630

-------------------------------------
SUBROUTINE 5000
-------------------------------------
THIS SUBROUTINE GATHERS THE DATA ENTERED ON THE STUDENT FORM.
LOCATX CONTAINS ROW, COLUMN (COL), THE NUMBER OF CHARACTERS TO BE READ (NCHARS), AND ANY COLUMNS TO SKIP WHILE PICKING UP THE DATA (NSKIP1 & NSKIP2).
INFO(1 AND 2) HOLD THE INFORMATION READ IN.

P1 POINTS TO THE LOCATION IN INFO* BEING USED. P2 POINTS TO THE CHARACTER NUMBER.

INFO*(PAGENO) = ""
THIS INSURES INFO* IS EMPTY
FIRST TIME THROUGH LOCATX---JUST MOVING CURSOR
LOCATE ROW, COL
INPUT "",B\$
IF B\$ = "MENU$" OR B\$ = "menu$" THEN FLAG = 1: RETURN
GOTO 5210
REM THE SECOND TIME THROUGH LOCATX PICKS UP
REM INFORMATION FROM THE SCREEN FOR INFO$
P1 = 1
ROW = LOCATX(P1)
P1 = P1 + 1
COL = LOCATX(P1)
P1 = P1 + 1
NCHARS = LOCATX(P1)
P1 = P1 + 1
NSKIPI = LOCATX(P1)
P1 = P1 + 1
NSKIP2 = LOCATX(P1)
IF NSKIP2 = -1 THEN RETURN
TEMPINFO$ = ""
P1 = P1 + 1
LOCATE ROW,COL
TEMPINFO$ = TEMPINFO$ + CHR$(SCREEN(ROW,COL))
COL = COL + 1
IF COL = NSKIP1 OR COL = NSKIP2 THEN COL = COL + 1
P2 = P2 + 1
IF P2 <= NCHARS THEN
LOCATE ROW,COL
PRINT MID$(INFO$(PAGENO),P2,1);
X = ASC(TEMPINFO$)
IF X <> 32 THEN 5610
NCHAR = NCHAR - 1
MS = MID$(TEMPINFO$,2)
TEMPINFO$ = MS + " "
IF NCHAR <= 0 THEN 5610
INFO$(PAGENO) = INFO$(PAGENO) + TEMPINFO$
GOTO 5350
--------------------------------------------------------
SUBROUTINE 6000
--------------------------------------------------------
THIS SUBROUTINE PUTS DATA FROM INFO$(1 OR 2) ONTO THE STUDENT FORM.
Variables used are the same as the previous subroutine.
6250 P2 = P2 + 1
6260 NCHARS = NCHARS - 1
6270 IF NCHARS < 1 THEN 6120
6280 COL = COL + 1
6290 IF COL = NSKIP1 OR COL = NSKIP2 THEN PRINT "-"; COL = COL + 1
6300 GOTO 6240
0k
APPENDIX D

UPDAY--INTERVIEW FILE CREATION PROGRAM

This appendix presents a functional description and complete listing of UPDAY. The use of UPDAY is discussed in Chapter 5. In order to understand this section better, it would be best to read the discussion in Chapter 5 first.

Functional Description

UPDAY maintains the interview schedules. At the beginning of each interview day, a staff person runs UPDAY in order to open interview files for the prospective employers who are within 14 days of their first date, to close (delete) the interview files for employers who are done interviewing, and to delete employer files also, if that is desired.

In lines 320 and 330, the maximum number of employer and interview files which will be examined is set. These numbers will need changing when more storage space is available.

Lines 340-420 describe the use of UPDAY. Since there is no menu, lines 430-460 are used to allow the user to pause while reading the description and to choose to continue or return to EXECUTIV. Following that, the business of UPDAY can commence.
In lines 480-500, the first employer cross file is gotten and tested in the following ways: Lines 530 and 540 determine if it is all blanks or zeros, and if so the next cross file is loaded. Lines 560 and 570 determine the numerical value (in days) of the first and last interview dates, so that they can be compared with the numerical value of today's date determined in line 580. Line 590 checks to determine if the first interview date is within 14 days of today; if so, interviews should be opened. Line 640 checks to see if all interview dates are in the past; if so, the interview files should be deleted and perhaps the employer file should also be deleted.

Subroutine "Print cross reference" is called to display the name, division, and first and last interview dates for the employer whose file has been called up. It also tests the "interviews opened" flag and reports whether interviews have been previously opened or not.

Lines 680-750 direct program execution to the subroutine which opens interview files, if the user agrees. Lines 760-750 handle the deletion of interview files, and lines 870-930 delete the employer file, both under the direction of the system user.

Upon completion of interview file work on the previous employer, line 940 directs the program execution
back to line 500 in order to look at the next file. Lines 950-1060 announce program completion after all files have been examined (as determined in lines 510 and 520--either a file with all null characters is encountered or MAXEMP is reached.)

In UPDAY, the main program opens EMPLCROS and holds it open for its entire operating time. Subroutine "Print cross reference" uses the information gotten from the cross file and held in CROSREF$ to display the employer name, division, etc. Its operation is straightforward. Use of the Basic function MID$ requires the string variable name, then the first character number, and finally the number of characters desired.

Subroutine YESNO returns a one for a "yes" answer, a two for a "no" answer, and will not accept anything else. This subroutine, in one form or another, is used in all the programs described in these appendices.

Subroutine "Setup interview files" opens the interview files. The larger loop, starting at line 1710, opens interview files until less than one interviewer is encountered in line 1820. The loop beginning at line 2000 and a second beginning at 2060 count the number of interviewers, while the innermost loops (lines 2010 and 2070) count out the number of morning and afternoon interviews. The logic of the interview files is fairly
in INFO$ (1-3 currently). The two little loops at lines 2380 and 2420 place the number of interviewers available in each of the interview slots available in the employer schedule. Refer to the FILETEST printout of an employer file in Figure 36 of Chapter 7: The third page of the EMPLYRS file gives a list of all the interviewers still available. When the interviews are initially set up the interviewers still available is just the total number of interviewers, and that number is written into each time slot for which interviews will be scheduled. The purpose of this duplication of information is to allow the various programs to operate more rapidly--there is no need to open interview files unless an interview is actually going to be scheduled.

Lines 2510 and 2520 complete the larger loop, returning execution to line 1720 in order to do the next interview day. Lines 2610-2630 give the file number for the employer whose interview files were just opened, and lines 2660-2700 complete the filing of information. File one is the employer cross file, which is held open, as it is closed in the main program.

Subroutine "Scrap interview files" deletes interview files. First it gets the employer file (lines 2890-2910). The loop starting at line 2920 gathers up to five interview file numbers (SCHEDNO$(1 to 5)), then lines 3030 and 3040
get the first file. The large loop starts with NDEX equal
to one at line 3020 and is completed at line 3280. Lines
3060-3100 allow the option of looking at each of the files
before deletion in case some item of information within the
file needs to be recorded. This print routine is
relatively straightforward and will not be discussed. Line
3310 gives the last opportunity to abort the erasure of all
the interview files. If the go-ahead is given, lines
3350-3380 write zeros into the three variables holding
interview files, and lines 3390-3430 actually erase the old
files.

Subroutine "Scrap employer file" deletes the employer
file. Line 3660 gives the option of a listing of the
employer file first. Lines 3700-3900 actually print the
file. Lines 3910-3930 give a final chance to abort the
deletion of the employer file, and if the go-ahead is
given, lines 3950-4030 do the actual deletion. The logic
used in this subroutine is nearly identical to that of
"Scrap interview files," so detailed description is not
necessary.
UPDAY Listing

LIST
10 REM UPDAY.BAS WRITTEN 11-11-84 JOHN C. DEBO
20 REM REVISED AND ADDED SUBROUTINE 4000 11-20-84
30 REM CHANGES 12-2-84
40 REM CORRECTED ERRORS IN 4000 1-6-84
50 REM CORRECTED CORRECTIONS ON 2-2-85
60 REM RENUMBERED 2-16-85
70 REM
80 REM
90 REM UPDAY IS USED BY A STAFF PERSON TO GO THROUGH
100 REM ALL THE EMPLOYER FILES CHECKING FOR THOSE WHICH
110 REM NEED TO HAVE INTERVIEW FILES OPENED. THIS IS
120 REM AN INTERACTIVE PROCESS TO ALLOW STAFF DECISION
130 REM ON EACH EMPLOYER. A SECOND FUNCTION IS TO REMOVE
140 REM OLD INTERVIEW FILES AND EMPLOYER FILES AFTER
150 REM THEY'RE NO LONGER USEFUL.
160 REM
170 REM
180 REM INFO$(1 THRU 6) IS USED TO HOLD EMPLOYER AND
190 REM INTERVIEW FILE DATA WHILE IT IS PROCESSED.
200 REM
210 REM MAXEMP AND MAXINTV CONTAIN THE ALLOWABLE SIZE
220 REM OF EMPLOYER AND INTERVIEW FILES. THESE SHOULD
230 REM BE ADJUSTED TO REFLECT THE SIZE OF THE STORAGE
240 REM MEDIUM AVAILABLE.
250 REM
260 REM CROSREF$ CONTAINS THE DATA FROM THE EMPLOYER
270 REM CROSS FILE POINTED TO BY FILENO.
280 REM
290 REM YESNO IS RETURNED BY SUBROUTINE 2500 (1 = YES,
300 REM 2 = NO).
310 REM
320 MAXEMP = 100
330 MAXINTV = 100
340 CLS
350 PRINT:PRINT:PRINT
360 PRINT " THIS PROGRAM IS 'UPDAY'."
370 PRINT " IT WILL HELP YOU OPEN INTERVIEWS FOR UPCOMING" EMPLERs.
380 PRINT " YOU WILL NEED TO DECIDE WHETHER TO"
390 PRINT " OPEN EACH EMPLOYER'S INTERVIEW SCHEDULE AS IT"
400 PRINT " IS CALLED OUT. YOU WILL ALSO BE ASKED WHETHER"
410 PRINT " TO DELETE OLD INTERVIEW AND EMPLOYER FILES."
420 PRINT " DO YOU WISH TO CONTINUE? (Y/N)"
430 PRINT "
440 FLAG = 0
450 GOSUB 1300
460 ON YESNO GOTO 480,470
470 RUN "A:EXECUTIV.BAS"
480 OPEN "R", 1, "B:EMPLCROS",64
490 FIELD 1, 64 AS CROSREF$
500 LET 1
510 IF LOC$(1) = MAXEMP + 1 THEN 950
520 IF CROSREF$ = STRING$(64,0) THEN 950
530 IF LEFT$(CROSREF$,62) = STRING$(62,48) THEN 500
540 IF LEFT$(CROSREF$, 62) = STRING$(62, 32) THEN 500
550 FILENO = LOC(1)
560 FIRSTDATE = VAL(MID$(CROSREF$, 55, 2)) * 365 + VAL(MID$(CROSREF$, 51, 2)) * 30 + VAL(MID$(CROSREF$, 53, 2))
570 LASTDATE = VAL(MID$(CROSREF$, 61, 2)) * 365 + VAL(MID$(CROSREF$, 57, 2)) * 30 + VAL(MID$(CROSREF$, 59, 2))
580 TODAY = VAL(MID$(DATE$, 9, 2)) * 365 + VAL(MID$(DATE$, 1, 2)) * 30 + VAL(MID$(DATE$, 4, 2))
590 IF TODAY + 14 < FIRSTDATE THEN 500
600 REM
610 REM INSERT THE NUMBER OF DAYS LEAD TIME--
620 REM I.E. HOW FAR IN ADVANCE TO OPEN INTERVIEWS
630 REM
640 IF TODAY > LASTDATE THEN 760
650 IF MID$(CROSREF$, 63, 1) = "1" THEN 500
660 GOSUB 1070 'PRINTS THE EMPLOYER'S NAME, ETC.
670 PRINT:
680 PRINT " >>> DO YOU WISH TO OPEN INTERVIEWS? (Y/N)"
690 PRINT "----------------------------------------
700 GOSUB 1300
710 ON YESNO GOTO 720, 500
720 GOSUB 1460 'OPENS THE INTERVIEW FILES
730 IF FLAG = 1 THEN CLOSE 1; GOTO 340
740 PUT 1, FILENO 'SET "INTERVIEWS SCHEDULED" FLAG
750 GOTO 500
760 REM ----------------------------------------
770 REM CHECK DELETE--NOT A SUBROUTINE
780 REM
790 GOSUB 1070
800 PRINT:
810 PRINT " >>> DO YOU WISH TO DELETE THE INTERVIEW FILES FOR THIS EMPLOYER? (Y/N)"
820 PRINT "----------------------------------------
830 GOSUB 1300
840 ON YESNO GOTO 850, 500
850 GOSUB 2770
860 PUT 1, FILENO 'CLEAR "INTERVIEWS SCHEDULED" FLAG
870 GOSUB 1070
880 PRINT:
890 PRINT " >>> DO YOU ALSO WISH TO ERASE THIS EMPLOYER FILE? (Y/N)"
900 PRINT "----------------------------------------
910 GOSUB 1300
920 ON YESNO GOTO 930, 500
930 GOSUB 3520
940 GOTO 500
950 REM ----------------------------------------
960 REM WRAP IT UP--NOT A SUBROUTINE
970 REM
980 GOSUB 1300
990 CLOSE 1
1000 PRINT: PRINT: PRINT " UPDAY IS COMPLETE--ALL FILES HAVE BEEN EXAMINED."
1010 PRINT " >>> DO YOU WISH TO RE-RUN THIS PROGRAM TO MAKE ANY MORE CHANGES? (Y/N)"
1020 GOSUB 1300
1030 GOSUB 1300
1040 CLOSE 1
1050 ON YESNO GOTO 1060, 1060
1060 RUN "A:EXECUTIV.BAS"
1070 REM ----------------------------------------
1080 REM  SUBROUTINE  PRINT CROSS REFERENCE
1090 REM  ---------------------------------------------
1100 REM  THIS SUBROUTINE PRINTS THE CROSREFERENCE
1110 REM  INFORMATION ABOUT THE EMPLOYER (NAME,DIVISION
1120 REM  FIRST & LAST INTERVIEW DATES, AND WHETHER
1130 REM  INTERVIEWS HAVE BEEN OPENED (CHARACTER 63 OF
1140 REM  EMPLCROS = 1).
1150 REM
1160 CLS
1170 PRINT:PRINT:PRINT
1180 PRINT "EMPLOYER FILE NUMBER ";FILENO;" IS SHOWN BELOW"
1190 PRINT:PRINT
1200 PRINT " NAME: ",MIDS(CROSREFS,1,25)
1210 PRINT " DIVISION: ",MIDS(CROSREFS,26,25)
1220 PRINT " FIRST INTERVIEW DATE",MIDS(CROSREFS,51,2);"-";MIDS(CROSREFS,53,2);" - ";MIDS(CROSREFS,55,2)
1230 PRINT " LAST INTERVIEW DATE",MIDS(CROSREFS,57,2);"-";MIDS(CROSREFS,59,2);" - ";MIDS(CROSREFS,61,2)
1240 IF MIDS(CROSREFS,63,1) = 1 THEN PRINT "INTERVIEWS WERE PREVIOUSLY OPENED"
1250 ELSE PRINT "INTERVIEWS HAVE NOT BEEN OPENED"
1260 RETURN
1270 REM  ---------------------------------------------
1280 REM  SUBROUTINE YESNO
1290 REM  -------------------------------
1300 REM  THIS SUBROUTINE HANDLES YES/NO
1310 REM  QUESTIONS, RETURNING YESNO = 1 FOR
1320 REM  YES AND YESNO = 2 FOR NO
1330 REM
1340 SOUND 523,3
1350 NS=""
1360 WHILE LEN(NS) < 1
1370 NS = INKEY$
1380 WEND
1390 IF NS = "N" OR NS = "n" THEN YESNO = 2:RETURN
1400 IF NS = "Y" OR NS = "y" THEN YESNO = 1:RETURN
1410 SOUND 100,4
1420 GOTO 1380
1430 REM  -------------------------------
1440 REM  SUBROUTINE SETUP INTERVIEW FILES
1450 REM
1460 REM  THIS SUBROUTINE OPENS THE INTERVIEW
1470 REM  FILES SO STUDENTS CAN SIGN UP. UP TO
1480 REM  5 FILES WILL BE OPENED, ONE FOR
1490 REM  EACH INTERVIEW DATE.
1500 REM
1510 REM  THE LARGE LOOP OPENS INTERVIEW FILES UNTIL
1520 REM  IT ENCOUNTERS <1 INTERVIEWER ON A DATE LINE.
1530 REM
1540 OPEN "R",2, "B:EMPLVRS", 512
1550 FIELD 2, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
1560 GET 2, FILENO
1650 INFO$1 = PG1INFO$  
1660 INFO$2 = PG2INFO$  
1670 INFO$3 = STRING$(130,48)  
1680 REM  
1690 REM THIS LOOP REPEATS FOR EACH INTERVIEW DATE  
1700 REM  
1710 N = 1  
1720 P = N$9 + 115  'POINTS TO INTERVIEW DATE IN EMPLOYER  
1730 N$ = " "  
1740 RSET N$ = STRING$(FILENO)  
1750 INFO$(6) = MID$(INFO$(2),P,9) + MID$(INFO$(2),169,9) + N$  
1760 REM  
1770 REM THIS CREATES THE THIRD PAGE OF THE SCHEDULE  
1780 REM FILE. THE DATA IS IDENTICAL TO EMPLYRS FILE--  
1790 REM IT ALSO INCLUDES THE EMPLOYER'S FILE NUMBER.  
1800 REM  
1810 NOINTVRS = VAL(MIDS(INFO$(2),1))  
1820 IF NOINTVRS < 1 OR NOINTVRS > 5 THEN 2610 'TIME TO QUIT  
1830 IF NOINTVRS <= 6 THEN 1930  
1840 CLS:PRINT:PRINT:PRINT  
1850 PRINT " THERE IS AN ERROR IN THE EMPLOYER FILE -- IT MAY BE ASKING"  
1860 PRINT " FOR MORE THAN 6 INTERVIEWERS ON A GIVEN DATE, OR SOMTHING"  
1870 PRINT " SIMILAR. CORRECT THE EMPLOYER FILE AND THEN RERUN UPDAY."  
1880 FOR N = 1 TO 2000  
1890 NEXT N  
1900 FLAG = 1  'ABORT THE OPERATION  
1910 CLOSE 2  
1920 RETURN  
1930 AMINTVS = VAL(MIDS(INFO$(2),P+7,1))  
1940 IF AMINTVS < 0 OR AMINTVS > 9 THEN 1840  
1950 PMINTVS = VAL(MIDS(INFO$(2),P+8,1))  
1960 IF PMINTVS < 0 OR PMINTVS > 9 THEN 1840  
1970 INFO$(4) = STRING$(216,48)  
1980 INFO$(5) = STRING$(216,48)  
1990 IF AMINTVS = 0 THEN 2060  
2000 FOR A = 1 TO NOINTVRS  
2010 FOR B = 0 TO AMINTVS - 1  
2020 MIDS(INFO$(4),4*A-3+24*B,4) = STRING$(4,32)  
2030 NEXT B  
2040 NEXT A  
2050 IF PMINTVS = 0 THEN 2110  
2060 FOR A = 1 TO NOINTVRS  
2070 FOR B = 0 TO PMINTVS - 1  
2080 MIDS(INFO$(5),4*A-3+24*B,4) = STRING$(4,32)  
2090 NEXT B  
2100 NEXT A  
2110 REM  
2120 REM THESE LOOPS HAVE PUT O'S IN ALL INTERVIEW  
2130 REM SLOTS (INDICATING CLOSED); THEN WRITTEN  
2140 REM SPACES IN THOSE WHICH REPRESENT ACTUAL  
2150 REM INTERVIEW OPPORTUNITIES (INDICATING OPEN).  
2160 REM NOW FIND AN OPEN INTERVIEW FILE.  
2170 REM  
2180 INTFILE = 1  
2190 OPEN "R",3, "B:EMPLSCHED",512  
2200 FIELD 3, 216 AS PG4INFO$, 216 AS PG5INFO$, 80 AS PG6INFO$  
2210 GET 3,INTFILE  
2220 IF INTFILE > MAXINTV THEN 2530 'NO FILES AVAILABLE!  
2230 IF PG4INFO$ <> STRING$(216,0) AND PG4INFO$ <> STRING$(216,48) THEN INTFILE  
2240 = INTFILE + 1; GOTO 2210
THIS LOOP REPEATS FOR EACH INTERVIEW DATE

POINTER TO INTERVIEW DATE IN EMPLOYER

THIS CREATES THE THIRD PAGE OF THE SCHEDULE
FILE. THE DATA IS IDENTICAL TO EMPLOYER FILE--
IT ALSO INCLUDES THE EMPLOYER’S FILE NUMBER.

NOINTVRS = VAL(MIDS(INFOS(2),P+7,1))

TIME TO QUIT

ABORT THE OPERATION

OPEN "R",3, "B:EMPLSCHED",512

FILED 3, 216 AS PG4INFO$, 216 AS PG5INFO$, 80 AS PG6INFO$

GET 3,INTFILE

NO FILES AVAILABLE!

NOW FIND AN OPEN INTERVIEW FILE.
N will contain the number of files.
NEX points to the file being operated on.
Val(SchedNOS(NDEX)) gives the interview file number.

Open "R", 2, "B:EMPLYS", 512
Field 2, 205 as PG1INFO$, 177 as PG2INFO$, 130 as PG3INFO$

Get 2, FILEN

For N = 1 to 5

SchedNOS(N) = Mid$(PG3INFO$, P, 4)
If SchedNOS(N) = String$(4, 0) or SchedNOS(N) = String$(4, 48) or SchedNOS(N) = String$(4, 32) then 2970

Next N

N = N - 1

Now we have the file numbers for the interview files associated with the employer file. Get the first one.

NDEX = 1
Open "R", 3, "B:EMPLSCHED", 512
Field 3, 216 as PG4INFO$, 216 as PG5INFO$, 80 as PG6INFO$

Cls
Print Print Print
Print "Do you wish to look at employer number "; FILEN
Print "Print interview schedules before deleting them? (Y/N)"
Gosub 1300

On YESNO Goto 3110, 3300
This = Val(SchedNOS(NDEX))
If This < 1 or This > MaxEMP Then 3270

Get 3, This
Cls
Print Print Print
Print "This is schedule file number "; NDEX; " A.M. SCHEDULE --- P.M. SCHEDULE ---"
Print "Interviewer:"
Print "1 2 3 4 5 6 1 2 3 4 5 6"
For M = 1 to 9

P = 24*M - 23
Mid$(PG4INFO$, P, 24); "; Mid$(PG5INFO$, P, 24)
Next M

Cls
Print Print Print
Print " << Press 'Y' when you're ready to continue >> "
Gosub 1300

NDEX = NDEX + 1
If NDEX > N Then 3290 Else 3110
Cls

Cls
Print Print Print
Print "Do you still wish to erase the interview schedules? (Y/N)"
Print " "
Gosub 1300
On YESNO Goto 3350, 3460
N$ = String$(216, 48)
LSet PG4INFO$ = N$
LSet PG5INFO$ = N$
LSet PG6INFO$ = N$
For NDEX = 1 to N
This = Val(SchedNOS(NDEX))
If This < 1 or This > MaxEMP Then 3430
Put 3, This
Gosub 3430

131
3440 LSET PG3INFO$ = STRING$(130,0)
3450 PUT 2,FILENO
3460 CLOSE 3
3470 CLOSE 2
3480 NS=CROSREF$
3490 MID$(NS,63,1) = "0"
3500 LSET CROSREF$ = NS
3510 RETURN
3520 REM SUBROUTINE SCRAP EMPLOYER FILE
3530 REM THIS SUBROUTINE DELETES THE EMPLOYER
3540 REM FILE. IT OFFERS THE OPTION OF A
3550 REM PRINTED COPY OF THE FILE BEFORE
3560 REM IT IS DESTROYED
3570 REM
3580 OPEN "R",2, "B:EMPLOYRS", 512
3590 FIELD 2, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
3600 GET 2, FILENO
3610 INFO$(1) = PG1INFO$
3620 INFO$(2) = PG2INFO$
3630 PRINT:PRINT
3640 PRINT "DO YOU WISH TO LOOK AT THE EMPLOYER FILE TO BE DELETED? (Y/N)"
3650 GOSUB 1300
3660 ON YESNO GOTO 3690, 3910
3670 GOSUB 1300
3680 ON YESNO GOTO 3690, 3910
3690 REM
3700 PRINT "YOU MAY USE <PRTSC> TO MAKE A COPY OF THIS INFORMATION"
3710 PRINT "EMPLOYER FILE NUMBER "FILENO;" IS SHOWN BELOW:"
3720 PRINT
3730 PRINT MID$(INFO$(1),1,25)
3740 PRINT MID$(INFO$(1),26,25)
3750 PRINT MID$(INFO$(1),51,25)
3760 PRINT MID$(INFO$(1),76,15); ""MID$(INFO$(1),91,2)
3770 PRINT MID$(INFO$(1),93,9); ""MID$(INFO$(1),102,10).
3780 PRINT MID$(INFO$(1),112,15); ""MID$(INFO$(1),127,14); ""MID$(INFO$(1),141,1)
3790 PRINT MID$(INFO$(1),142,25)
3800 PRINT MID$(INFO$(1),167,10); ""MID$(INFO$(1),177,7)
3810 PRINT MID$(INFO$(1),184,10)
3820 PRINT MID$(INFO$(1),194,12)
3830 PRINT
3840 PRINT MID$(INFO$(2),1,40)
3850 PRINT MID$(INFO$(2),41,4); ""MID$(INFO$(2),45,1); ""MID$(INFO$(2),46,25)
3860 PRINT MID$(INFO$(2),71,3); ""MID$(INFO$(2),74,50)
3870 PRINT MID$(INFO$(2),124,9); ""MID$(INFO$(2),133,9); ""MID$(INFO$(2),142,9)
3880 PRINT MID$(INFO$(2),151,9); ""MID$(INFO$(2),160,9); ""MID$(INFO$(2),169,1)
3890 PRINT MID$(INFO$(2),170,4); ""MID$(INFO$(2),174,4)
3900 PRINT
3910 PRINT ">>> ARE YOU SURE THIS EMPLOYER FILE SHOULD BE ERASED? (Y/N)"
3920 GOSUB 1300
3930 ON YESNO GOTO 3950, 3940
3940 GOSUB 1300
3950 INFO$(1) = STRING$(205,48)
3960 INFO$(2) = STRING$(177,48)
3970 INFO$(3) = STRING$(130,48)
3980 LSET CROSREF$ = STRING$(64,48)
3990 PUT 1,FILENO
4000 LSET PG1INFO* = INFO*(1)
4010 LSET PG2INFO* = INFO*(2)
4020 LSET PG3INFO* = INFO*(3)
4030 PUT 2,FILENO
4040 CLOSE 2
4050 RETURN
Ok
APPENDIX E

DAYSCHED--INTERVIEW SCHEDULE WRITING PROGRAM

This appendix presents a functional description and complete listing of DAYSCHED. In order to understand this section better, it would be best to read the discussion of the use of DAYSCHED in Chapter 6.

**Functional Description**

DAYSCHED can serve two functions: It can be used to list all of a particular day's appointments and it can list one or all scheduled days for an individual employer. The initial functions are similar to the preceding programs with the first menu option commencing at line 480 with a request for the date desired. This date is printed for confirmation, and then subroutine "Who's interviewing?" is called. This subroutine, discussed in Appendix A, returns the array INTFILE which contains INDEX number of employer file numbers for employers who may be interviewing on the date given. Line 610 then calls "Employer data" which returns the up to five interview dates for that company. The inner loop starting at line 620 checks all five dates to see if one of them is the desired date. If it is, lines 660-690 print the company name and division from two other
variables returned by the subroutine and line 720 calls subroutine "Print schedule" to print the schedule for that company, and loops back to 600 to look for more companies. This actually completes the main program's job for the first option; subroutines will be discussed shortly.

The second option, printing an individual company's schedule(s), starts at line 770 and uses very similar logic to that described for the first option.

Subroutine "Student data printer" opens a given student file, and obtains the name, social security number, telephone number, grade point, graduation date, major and citizenship employing logic almost identical to that used in STDNSRV for the same function.

Subroutine "Print schedule" opens the employer schedule and obtains the second and third pages of information (lines 2380-2460) which contains the interview schedule information. The outer loop (line 2530-2760) counts through the number of interviewers given, the next loop (lines 2560-2740) does morning and then afternoon times, and the inner loop (2590-2730) counts the interview time slots (one to a maximum of nine). Within these three loops the interview times are determined in lines 2610 and 2620 and printed in line 2680, then the "Student data printer" subroutine is called to put in the information about the student scheduled for that time.
Subroutines "Yesno" and "Choice" have been previously described, as has "Infoget." Subroutine "Get emplyr fileno" was described in Appendix B, EMPLUP.
LIST
10 REM DAYSCHED.BAS WRITTEN 2-8-85 JOHN C. DEBO
20 REM COMPLETE & TROUBLESHOOT THROUGH 2-16-85
30 REM
40 REM
50 REM
60 REM POSSIBLE MODIFICATION--LPRINTS IF PRINTER AVAILABLE
70 REM
80 REM DAYSCHED SERVES TWO FUNCTIONS:
90 REM 1) IT WILL LIST ALL
100 REM OF A PARTICULAR DAY'S APPOINTMENT SCHEDULES BY
110 REM COMPANY, INTERVIEWER NUMBER, AND TIME.
120 REM 2) IT WILL
130 REM LIST ONE DAY OR ALL OF AN INDIVIDUAL COMPANY'S
140 REM INTERVIEWS. IT GIVES STUDENT NAME, SSN, AND OTHER
150 REM USEFUL INFORMATION ON EACH INTERVIEW.
160 REM
170 REM INFO$(1 THRU 6) IS USED TO HOLD EMPLOYER AND
180 REM STUDENT FILE INFORMATION AND TO GET INTERVIEW
190 REM
200 REM MAXSTU, MAXEMP, AND MAXINTV CONTAIN THE ALLOWABLE
210 REM SIZE FOR THE STUDENT, EMPLOYER, AND INTERVIEW FILES.
220 REM THESE MUST BE SET TO REFLECT THE SIZE OF STORAGE
230 REM MEDIUM AVAILABLE.
240 REM CROSREF$ CONTAINS THE DATA FROM THE EMPLOYER
250 REM CROSS FILE POINTED TO BY FILENO.
260 REM ROW AND COL ALWAYS REFER TO SCREEN LOCATIONS.
270 REM
280 REM YESNO AND CHOICE ARE EXPLAINED IN THEIR SUBROUTINES.
290 REM
300 REM
310 MAXSTU = 100
320 MAXEMP = 100
330 MAXINTV = 100
340 DIM INTFILE(50) 'MAXIMUM NO. CHECKABLE INTERVIEWS
350 REM
360 CLS PRINT PRINT
370 PRINT " THIS PROGRAM PRINTS A SCHEDULE FOR ONE INTERVIEW DAY. IT"
380 PRINT " MAY ALSO BE USED TO PRINT ALL OF A PARTICULAR COMPANY'S SCHEDULES"
390 PRINT" DO YOU WISH TO:
400 PRINT" 1) PRINT A DAY'S SCHEDULE?"
410 PRINT" 2) PRINT A COMPANY'S SCHEDULE?"
420 PRINT" 3) RETURN TO THE MAIN MENU? (CHOOSE 1 TO 3)"
430 CHOICE = 3
440 GOSUB 2950
450 ON CHOICE GOTO 480,770,450
460 RUN "A:EXECUTIV.BAS"
470 REM -----------------------------
480 REM ROUTINE FOR DAY'S SCHEDULE
490 PRINT " ENTER THE DESIRED DATE BELOW AS SHOWN"
500 PRINT " 02-07-84 <CR>"
510 INPUT " DATE==">",NS
520 THEDATE$ = MID$(NS,1,2) + MID$(NS,4,2) + MID$(NS,7,2)
530 PRINT " IS THIS THE DATE YOU WANT: ";MID$(THEDATE$,1,2);"-";MID$(THEDATE$
540 PRINT" ? (Y/N)"
530 GOSUB 2780
540 ON YESNO GOTO 550,480
550 GOSUB 3350 'GET LIST OF POSSIBLE COMPANIES
560 IF INDEX <> 0 THEN 590
570 PRINT:PRINT:PRINT" THERE ARE NO COMPANIES INTERVIEWING ON THAT DATE":PRINT
580 GOTO 390
590 INTCOUNT = INDEX
600 FOR INDEX = 1 TO INTCOUNT 'CHECK ALL THE POSSIBLE COMPANIES
610 GOSUB 1980
620 FOR N1 = 1 TO N 'CHECK ALL THE DATES IN COMPANY FILE
630 IF INTVDATE*(N1) = THEDATE* THEN 660
640 NEXT N1
650 GOTO 710 'NEXT INDEX--NO LUCK THAT COMPANY
660 PRINT:PRINT STRING$(78,205):PRINT
670 PRINT "COMPANY NAME: ";EMPNAME*
680 PRINT " DIVISION: ";DVSN*
690 PRINT
700 SCHDFILENO = VAL(INTVFILE*(N1))
710 REM
720 GOSUB 2380 'PRINT THAT COMPANY'S SCHEDULE
730 NEXT INDEX
740 PRINT:PRINT STRING$(78,205)
750 PRINT:PRINT" THAT'S ALL THE INTERVIEWING COMPANIES LISTED.":PRINT
760 GOTO 390
770 REM -----------------------------
780 REM ROUTINE FOR ONE COMPANY PRINT
790 REM
800 GOSUB 3590 'GET THE COMPANY FILE NUMBER
810 IF FLAG = 1 THEN 390
820 INDEX = 1:INTFILE(1) = FILENO
830 GOSUB 1980 'GET COMPANY'S INTERVIEW FILE
840 IF N > 0 THEN 870
850 PRINT:PRINT"THERE ARE NO INTERVIEW SCHEDULES IN THE COMPANY FILE":PRINT
860 GOTO 390
870 PRINT:PRINT" INTERVIEWS ARE SCHEDULED ON THESE DATES: ":PRINT
880 FOR P = 1 TO N
890 PRINT Pj") "LEFT((INTVDATE*(P),2));""MID((INTVDATES(P),3,2));""MID((I
900 NEXT P
910 PRINT:PRINT"DO YOU WISH A PRINT FOR ALL OF THE DATES ABOVE? (Y/N)"
920 GOSUB 2780
930 ON YESNO GOTO 1100,960
940 REM ------------------------
950 REM ROUTINE FOR ONE DATE
960 PRINT:PRINT"WHICH ONE SHALL I PRINT? (1 TO ";Nj")
970 CHOICE = N
980 GOSUB 2950
990 SCHDFILENO = VAL(INTVFILE*(CHOICE))
1000 PRINT:PRINT STRING$(78,205):PRINT
1010 PRINT "COMPANY NAME: ";EMPNAME*
1020 PRINT Pj") "LEFT((INTVDATES(P),2));""MID((INTVDATES(P),3,2));""MID((I
1030 PRINT " DIVISION: ";DVSN*
1040 PRINT
1050 GOSUB 2380 'PRINT SCHEDULE THAT DAY
1060 PRINT
1070 GOTO 390
1080 REM ------------------------
1090 REM ROUTINE FOR ALL N DATES
1100 PRINT:PRINT STRING$(78,205):PRINT
1110 PRINT " COMPANY NAME: "; \EMPNAME;" " DIVISION: "; \DVSN;" 
1120 PRINT 
1130 FOR P = 1 TO N 
1140 PRINT STRINGS(78,196) 
1150 PRINT "DATE: "; \LEFT(INTVDATE(P),2);""; \MID(INTVDATE(P),3,2);""; 
MID(INTVDATE(P),5) 
1160 SCHEDFILENO = VAL(INTVFILE(P)) 
1170 GOSUB 2380 "PRINT SCHEDULE THAT DAY 
1180 NEXT P 
1190 PRINT" THAT'S ALL THE COMPANY'S INTERVIEWS LISTED."; \PRINT 
1200 GOTO 390 
1210 REM ----------------------------- 
1220 REM SUBROUTINE "STUDENT DATA PRINTER" 
1230 REM ----------------------------- 
1240 REM PASS IT STDFILENO AND IT PRINTS THE 
1250 REM STUDENT'S NAME, SSN, PHONE, ETC. 
1260 REM 
1270 REM 
1280 OPEN "R", 1, "B\STUDENTS", 256 
1290 FIELD 1, 154 AS PG1INFO$, 102 AS PG2INFO$ 
1300 GET 1, STDFILENO 
1310 CLOSE 1 
1320 INFOS(1) = PG1INFO$ 
1330 INFOS(2) = PG2INFO$ 
1340 REM NOTE THAT INFOS(12) CONTAIN STUDENT DATA NOW. 
1350 REM ------- 
1360 FIRSTCHAR = 1; \NOCHAR = 15; INFONO = 1 
1370 GOSUB 3200 
1380 LASTNAME$ = INFOGET$ 
1390 FIRSTCHAR = 16; \NOCHAR = 14 
1400 GOSUB 3200 
1410 FIRSTNAME$ = INFOGET$ 
1420 MIDS$ = MID$(INFOS(1),30,1) 
1430 SSNCODE$ = MID$(INFOS(1),31,3) + "" + MID$(INFOS(1),34,2) + "" + MID$(INFOS(1),36,4) 
1440 PRINT "STUDENT NAME: "; \LASTNAME;", "; \FIRSTNAME;", "; \MIDS;"." 
1450 PRINT " LOCAL TELEPHONE: " 
1460 REM ----------------------------- 
1470 REM 
1480 PRINT " FIRSTCHAR = 40; \NOCHAR = 3 
1490 GOSUB 3200 
1500 PRINT INFOGET$;""; 
1510 PRINT INFOGET$;""; 
1520 FIRSTCHAR = 43; \NOCHAR = 3 
1530 GOSUB 3200 
1540 PRINT INFOGET$;""; 
1550 FIRSTCHAR = 46; \NOCHAR = 4 
1560 GOSUB 3200 
1570 PRINT INFOGET$ 
1580 REM ----------------------------- 
1590 REM THIS SECTION DISPLAYS THE STUDENT'S G.P.A., 
1600 REM GRADUATION DATE, MAJOR, AND CITIZENSHIP. 
1610 REM 
1620 YEAR$ = MID$(INFOS(2),2,2) 
1630 IF VAL(YEAR$) < 84 THEN YEAR$ = "??" 
1640 NS$ = MID$(INFOS(2),1,1) 
1650 TERM$ = "UNKNOWN" 
1660 IF NS$ = "1" THEN TERM$ = "FALL" 
1670 IF NS$ = "2" THEN TERM$ = "SPRING"
1680 IF NS = "3" THEN TERMS = "SUMMER"
1690 PRINT "EXPECTED GRADUATION: " TERMS; " TERM OF " YEAR;
1700 REM --------------------
1710 PRINT "MAJOR CODE: " MID$(INFO$(2),4,4);
1720 REM
1730 CITIZ = "UNKNOWN STATUS"
1740 IF NS = "1" THEN CITIZ = "U.S. CITIZEN"
1750 IF NS = "2" THEN CITIZ = "PERMANENT VISA"
1760 IF NS = "3" THEN CITIZ = "WORK PERMIT"
1770 PRINT "CITIZENSHIP: "Citiz; "
1780 REM
1790 NS = MID$(INFO$(2),9,1)
1800 DEGR = "UNKNOWN"
1810 IF NS = "1" THEN DEGR = "BACHELORS"
1820 IF NS = "2" THEN DEGR = "MASTERS"
1830 IF NS = "3" THEN DEGR = "DOCTORATE"
1840 IF NS = "4" THEN DEGR = "SPECIAL CERTIFICATE."
1850 PRINT " DEGREE: " DEGR; ""
1860 REM -----------------------------
1870 YEAR = MID$(INFO$(2),14,2)
1880 IF VAL(YEAR) < 83 THEN YEAR = "??"
1890 NS = MID$(INFO$(2),13,1)
1900 TERMS = "UNKNOWN"
1910 IF NS = "1" THEN TERMS = "FALL"
1920 IF NS = "2" THEN TERMS = "SPRING"
1930 IF NS = "3" THEN TERMS = "SUMMER"
1940 GPA = VAL(MID$(INFO$(2),10,3)) / 100
1950 PRINT " THE STUDENT'S G.P.A. WAS: " GPA;
1960 RETURN
1970 REM
1980 REM -----------------------------
1990 REM SUBROUTINE "EMPLOYER DATA"
2000 REM -----------------------------
2010 REM CALL THIS ROUTINE WITH INTFILE<INDEX>
2020 REM POINTING TO POSSIBLE EMPLOYER. IT
2030 REM RETURNS EMPNAMES, DVSN, INTVDATE(1-5)
2040 REM AND INTVFILE(1-5). N = NUMBER OF
2050 REM INTERVIEWS IN THE TWO ARRAYS.
2060 REM
2070 OPEN "R",3,"B:EMPLVRS",512
2080 FIELD 3, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
2090 GET 3, INTFILE<INDEX>
2100 INFO$(1) = PG1INFO$
2110 INFO$(2) = PG2INFO$
2120 INFO$(3) = PG3INFO$
2130 CLOSE 3
2140 REM NOTE THAT INFO$(1-3) CONTAIN EMPLOYER DATA NOW.
2150 REM
2160 EMPNAMES = MID$(INFO$(1),1,25)
2170 DVSN = MID$(INFO$(1),26,25)
2180 FOR N = 1 TO 5 "THE FIVE INTERVIEWS POSSIBLE"
2190 INFDON = 2
2200 GOSUB 3200
2210 IF L = 0 THEN 2280
2220 IF L = 0 THEN 2280
2230 INFDON = 3
2240 GOSUB 3200
2250 INFTVFILE$(N) = INFOGET$
2260 IF L = 0 THEN 2280
SUBROUTINE PRINT SCHEDULE

CALL IT WITH SCHDFILENO, IT PRINTS
THE INTERVIEW NUMBER AND TIME, THEN
CALLS 'STUDENT DATA' TO PRINT THAT
INFORMATION.

OPEN "R",1,"B:EMPLSCHE", 512
FIELD 1, 216 AS PG1INFO$, 216 AS PG2INFO$, 80 AS PG3INFO$
GET 1,SCHDFILENO 'THE FILE NUMBER
INFO$(4) = PG1INFO$
INFO$(5) = PG2INFO$
INFO$(6) = PG3INFO$
NOTE THAT INFO$(4-6) CONTAIN SCHEDULE FILE DATA NOW.

OPEN .. R",1, .. B1EMPLSCHE", 512
FIELD 1, 216 AS PG1INFOS, 216 AS PG2INFOS, 80 AS PG3INFOS
GET 1,SCHDFILENO
INFOS(4) = PG1INFOS
INFOS(5) = PG2INFOS
INFOS(6) = PG3INFOS
REM NOTE THAT INFOS<4-6> CONTAIN SCHEDULE FILE DATA NOW.

CLOSE 1
NOINTVRS = VAL(MID$(INFO$(6),1,1))
AMINTVS = VAL(MID$(INFO$(6),8,1))
PHINTVS = VAL(MID$(INFO$(6),9,1))
LENGTH = VAL(MID$(INFO$(6),10,1))
AMSTART = VAL(MID$(INFO$(6),11,4))
PMSTART = VAL(MID$(INFO$(6),15,4))
FOR INTHR = 1 TO INTVS
INTERVAL = 15 + 15*LENGTH
MIN = 60 * FIX(START/100) + INTERVAL*(INTHR-1) + START - 100*FIX(START/100)
HRMN = FIX(MIN/60) & 100 + ((MIN/60)-FIX(MIN/60))%60
INFONO = MORNAFT + 31*NOCHAR = 4
FIRSTCHAR = 4*CINTVR - 1) + 24*(INTHR - 1) + 1
GOSUB 5300
STDFILENO = VAL(INFOGET$
IF STDFILENO < 1 OR STDFILENO > MAXSTU THEN 2730
PRINT;PRINT HRMN
PRNTCNT = PRNTCNT + 1
IF MORNAFT = 1 THEN PRINT " (AM) " ELSE PRINT " (PM) "
PRINT
GOSUB 1270
PRINT;PRINT HRMN
IF PRNTCNT = 0 THEN PRINT;PRINT " NO INTERVIEWS SCHEDULED ":PRINT
NEXT INTHR
NEXT MORNAFT
IF PRNTCNT = 0 THEN PRINT;PRINT " NO INTERVIEWS SCHEDULED ":PRINT
RETURN
SUBROUTINE YESNO
THIS SUBROUTINE HANDLES YES/NO
QUESTIONS, RETURNING YESNO = 1 FOR
YES AND YESNO = 2 FOR NO
SOUND 523,3
2860 NS = ""
2870 WHILE LEN(NS) < 1
2880 NS = INKEY$
2890 WEND
2900 IF NS = "N" OR NS = "n" THEN YESNO = 2:RETURN
2910 IF NS = "Y" OR NS = "y" THEN YESNO = 1:RETURN
2920 SOUND 100.4
2930 GOTO 2860
2940 REM SUBROUTINE CHOICE
2950 REM THIS SUBROUTINE HANDLES CHOICES OF
2960 REM MORE THAN ONE OPTION. CALL IT WITH
2970 REM CHOICE = NUMBER OF LARGEST OPTION; IT
2980 REM RETURNS WITH CHOICE = NUMBER OF OPTION
2990 REM CHOSEN.
3000 SOUND 523, 3
3010 IF N < 1 OR N > CHOICE THEN SOUNDF 100.4: GOTO 3040
3020 CHOICE = N
3030 RETURN
3040 REM SUBROUTINE INFOGETS
3050 REM THIS TAKES AN INFO$(INFONO) ARRAY, GETS
3060 REM A NOCHAR (NUMBER OF CHARACTERS) FROM
3070 REM IT STARTING AT FIRSTCHAR, THEN REMOVES
3080 REM ANY SPACES ON THE RIGHT END. RETURNS
3090 REM INFOGETS AND L (# CHAR. IN INFOGETS).
3100 INDEX = 1
3110 THEDAY = VAL(MIDS(THEDATE*, 5, 2)) + 10000 + VAL(MIDS(THEDATE*, 1, 2)) + 100 + VAL(MIDS(THEDATE*, 3, 2))
3120 GOTO 3190
3130 REM SUBROUTINE WHO'S INTERVIEWING?
3140 REM THIS SUBROUTINE CREATES A FILE CALLED
3150 REM 'INTFILE(INDEX)' WHICH CONTAINS THE FILE
3160 REM NUMBERS FOR EMPLOYERS WHO MAY BE INTERVIEW-
3170 REM ON THE DATE SPECIFIED IN 'THEDATE*'.
3180 REM SINCE THE EMPLOYER CROSS FILE CONTAINS
3190 REM ONLY FIRST AND LAST DATES, THE LIST ONLY
3200 REM CONTAINS POSSIBILITIES.
3210 INDEX = 1
3220 THEDAY = VAL(MIDS(THEDATE*, 5, 2)) + 10000 + VAL(MIDS(THEDATE*, 1, 2)) + 100 + VAL(MIDS(THEDATE*, 3, 2))
3230 OPEN "R", 1, "B:EMPLCROS", 64
3240 REM INDEX = 1
3250 GOTO 1
3260 REM
3270 REM
3440 IF LOC(1) = MAXEMP + 1 THEN 3560
3450 IF CROSREF$ = STRING$(64,0) THEN 3560
3460 IF LEFT$(CROSREF$, 62) = STRING$(62,48) THEN 3430
3470 IF LEFT$(CROSREF$, 62) = STRING$(62,32) THEN 3430
3480 FIRSTDATE = VAL(MID$(CROSREF$, 55,2)) * 10000 + VAL(MID$(CROSREF$, 51,2)) * 100
+ VAL(MID$(CROSREF$, 53,2))
3490 LASTDATE = VAL(MID$(CROSREF$, 61,2)) * 10000 + VAL(MID$(CROSREF$, 57,2)) * 100
+ VAL(MID$(CROSREF$, 59,2))
3500 IF THEDAY < FIRSTDATE THEN 3430
3510 IF THEDAY > LASTDATE THEN 3430
3520 IF MID$(CROSREF$, 63,1) <> "1" THEN 3430
3530 INFILE(INDEX) = LOC(1)
3540 INDEX = INDEX + 1
3550 GOTO 3430
3560 CLOSE 1
3570 INDEX = INDEX - 1
3580 RETURN

3590 REM ---------------------------------------------
3600 REM SUBROUTINE GET EMLVR FILENO
3610 REM ---------------------------------------------
3620 REM THIS SUBROUTINE FINDS THE PROPER
3630 REM EMLPYR FILE USING EITHER THE FILE
3640 REM NUMBER OR THE EMPLOYER NAME.
3650 REM
3660 FLAG = 0
3670 PRINT:PRINT"DO YOU KNOW THE EMPLOYER'S FILE NUMBER? (Y/N)"
3680 GOSUB 2780
3690 ON YESNO GOTO 3700, 3740
3700 PRINT:PRINT"ENTER THE FILE NUMBER, THEN <CR> PLEASE";
3710 SOUND 523, 3
3720 INPUT"===>", FILENO
3730 GOTO 4000
3740 PRINT:PRINT"ENTER EMPLOYER'S NAME AS IT APPEARS IN THE FILES, THEN <CR>"
3750 INPUT "EMPLOYER'S NAME===>", NAM$,
3760 NOCHAR = LEN(NAM$)
3770 OPEN "R", 1, "B:\EMPLCROS", 64
3780 FIELD 1, 64 AS CROSREF
3790 GET 1
3800 IF LOC(1) = MAXEMP + 1 THEN 3880
3810 IF CROSREF$ = STRING$(64,0) THEN 3880
3820 IF LEFT$(CROSREF$, NOCHAR) < NAM$ THEN 3790
3830 REM NUL CHARACTERS INDICATES VIRGIN TERRITORY
3840 REM --- NO NEED TO CONTINUE LOOKING.
3850 FILENO = LOC(1)
3860 GOTO 4000
3870 CLOSE 1
3880 PRINT "THERE IS NO EMPLOYER NAME IN THE FILE EXACTLY AS GIVEN."
3890 PRINT "--PERHAPS IT IS SPELLED DIFFERENT, ABBREVIATED, OR"
3900 PRINT "HAS A DIFFERENT NUMBER OF SPACES WITHIN IT?"
3910 PRINT
3920 PRINT
3930 SOUND 100, 4
3940 CLOSE 1
3950 PRINT "DO YOU WISH TO TRY AGAIN? (Y OR N)"
3960 GOSUB 2780
3970 ON YESNO GOTO 3670, 3980
3980 FLAG = 1
3990 REM WE HAVE A FILE NUMBER -- CHECK IT AGAINST
4000 REM THE CROSS FILE NOW TO BE SURE IT'S RIGHT.
4020 REM
4030 OPEN "R", 1, "B:EMPLCROS", 64
4040 FIELD 1, 64 AS CROSREF
4050 GET 1, FILENO
4060 PRINT: PRINT "THE SELECTED COMPANY IS SHOWN BELOW": PRINT
4070 PRINT "COMPANY: " ; MID$(CROSREF, 1, 25)
4080 PRINT "DIVISION: " ; MID$(CROSREF, 26, 25)
4090 PRINT: PRINT "IS IT CORRECT? (Y/N)"
4100 GOSUB 2780
4110 CLOSE 1
4120 ON YESNO GOTO 4130, 3950
4130 RETURN
Ok
APPENDIX F

PROGRAMS FOR STAFF USE

The staff used DINGLTR to print a letter to a student who has missed his or her interview appointment and MSGCREA to create the message files which communicate additional information to students who are signing up for an interview. Use of these programs was described in Chapter 6.

DINGLTR Functional Description

Lines 140-160 get the date missed, lines 190-210 the name of the interviewing company, and lines 240-210 get the student's file number. After the program checks whether the printer is on and ready, subroutine 2000 is called to get the information from the student file and print the letter heading. Then lines 360 through 740 print the body of the letter and upon completion, line 750 asks (on the screen) whether another letter is desired.

Line 340 and line 760 deal with the error trapping routine at line 1250-1300. The purpose is to avoid printer errors from stopping the program function. Commonly these errors might be failure to have the printer "On line" or not having paper in it. The statement in line 760 returns error handling to its normal mode.
LIST
10 REM DINGLTR.BAS NEW ON 11-17-84 JOHN C. DEBO
20 REM
30 REM
40 REM REVISIONS 11-20-84
50 REM REVISED 2-10-85
60 REM
70 CLS
100 PRINT "THIS PROGRAM WILL WRITE A LETTER TO A STUDENT WHO HAS MISSED A SCHEDULED" "INTERVIEW. PLEASE ENTER THE INFORMATION REQUESTED EXACTLY AS YOU WISH" "IT TO APPEAR. DO NOT USE ANY COMMAS IN THE DATA YOU ENTER!
120 PRINT "---
130 REM
140 PRINT "THE DATE THE STUDENT MISSED (LIKE THIS: 2-12-85)"
150 INPUT "ENTER DATE, THEN (CR) =="",MISDATE$)
160 IF LEN(MISDATE$) < 4 THEN SOUND 100,4:GOTO 140
170 REM
180 REM
190 PRINT "THE NAME OF THE INTERVIEWING COMPANY"
200 INPUT "COMPANY, THEN (CR) =="",MISCO$
210 IF LEN(MISCO$) < 5 THEN SOUND 100,4:GOTO 190
220 REM
230 REM
240 PRINT "THE STUDENT'S SYSTEM FILE NUMBER (1 TO 4 DIGITS)"
250 INPUT "FILE NUMBER, THEN (CR) ==",FILENO
260 IF FILENO < 1 OR FILENO > 9999 THEN 240
270 REM
280 REM
290 PRINT "FOR PRINTER ON AND AT FORM TOP? READY TO PRINT LETTER? (Y/N)"
300 PRINT "PRINTER ON AND AT FORM TOP? READY TO PRINT LETTER? (Y/N)"
310 GOSUB 800
320 ON YESNO GOTO 330,70
330 CLS
340 ON ERROR GOTO 1250
350 GOSUB 920
360 LPRINT "Our records indicate that you failed to fulfill your interview appointment on ",MISDATE$," for your interview with ",MISCO$,".
370 LPRINT "If you find that unavoidable circumstances prevent you from keeping your appointment, please extend the courtes of notifying the placement center."
380 LPRINT "Interview appointments must be fulfilled as scheduled or cancelled in advance to retain your interviewing privilege. If you fail to fulfill two interview appointments, it will be necessary to talk to a Placement Counselor prior to scheduling additional interviews."
390 LPRINT "It is important that you follow these procedures so that
we can be of maximum assistance to both you and the em-
ployers."

Should you have any questions, or wish to discuss other
services provided by the Placement Center, please come" in to see us."

Sincerely,

James W. Gracey"
Director"
Career Planning & Placemen
t
cc: department chairman

DO YOU WISH TO PRINT ANOTHER LETTER? (Y/N)"

SUBROUTINE 1000 -- "YESNO"

SUBROUTINE 2000

THIS SUBROUTINE OPENS THE STUDENT FILE

AND GETS THE NAME AND Mailing address.

OPEN "R",1,"B:STUDENTS",256
1000 FIELD 1, 154 AS PG1INFO$, 102 AS PG2INFO$
1010 GET 1,FILENO.

FIRSTNAME$ = MID$(PG1INFO$,16,14)
1020 LASTNAME$ = MID$(PG1INFO$,1,15)
1030 ADDRESS$ = MID$(PG1INFO$,61,20)
1040 CITY$ = MID$(PG1INFO$,81,15)
1050 STATE$ = MID$(PG1INFO$,96,2)
1060 ZIPCODE$ = MID$(PG1INFO$,98,5)
1070 CLOSE 1
1080 LPRINT "

";DATE$
1110 LPRINT:PRINT
1120 N = 14
1130 IF MID$(FIRSTNAME$,N,1) <> " " THEN 1160
1140 N=N-1
1150 GOTO 1130
1160 LPRINT"":MID$(FIRSTNAME$,1,N);"
1170 LPRINT"":LASTNAME$"
1180 LPRINT"":ADDRESS$"
1190 LPRINT"":CITY$"
1200 LPRINT"":STATE$":ZIPCODE$"
1210 LPRINT"
1220 LPRINT""Dear Student"
1230 LPRINT
1240 RETURN
1250 REM -------------------------------------
1260 REM ERROR HANDLING ROUTINE FOR PRINTER PROBS.
1270 REM
1280 PRINT:PRINT"":PLEASE CHECK THE PRINTER -- THERE IS A PROBLEM"
1290 PRINT
1300 RESUME 300
Ok
MSGCREA Functional Description

The message number is gotten in lines 240-270, and checked to see if it is within possible file range in line 280. The file is opened in line 290 and the message is checked to see if it exists. If not, line 350 starts the process of entering a new message. If a message exists, line 510 starts the process of printing and editing it. Subroutine "Yesno" has already been described, and the only unusual feature of this program is the subroutine "Input text lines."

Because Basic will not allow entry of character strings including commas, this subroutine was designed to duplicate most of the screen editing capabilities of Basic, but to accept any characters entered (up to 255) followed by a carriage return. The WHILE statement at line 940 and WEND at 1010 form a loop which puts a cursor at the row and column being edited and accepts a character when typed in. Line 1020 determines that a control character (left, right, up, or down arrow, etc.) was entered and transfers control to the part of the program for handling these editing functions starting at line 1140. If it was backspace or a carriage return, it will be trapped by line 1040 or 1050, otherwise it was a regular character which is printed in line 1070. Lines 1080 through 1110 handle the
logic of determining where the next character is to be printed (i.e., if it's at the end of a line it moves to the next line down, etc.).

Lines 1210 through 1300 determine which of the editing characters were typed and transfers control to the appropriate line below. If execution reaches line 1310 the character is one which is not used, it is ignored and the program returns to line 910 to input another character. Right arrow and left arrow merely increment or decrement the column number making sure the ends of the line are honored. Up arrow and down arrow decrement or increment the row number, making sure that the possible size of the message (255 characters) isn't exceeded. Control-left arrow and right arrow skip ahead or back to the previous word (looking for a space). Control-end takes the cursor to the right-hand edge of the page (column 80).

The insert and delete functions work somewhat differently (and more slowly) from their Basic counterparts. Insert puts in a space at the cursor location and does a sort of a bubble sort to move all the remaining characters down one in the string. Delete removes the character at the cursor and moves each character up in the string one at a time.
LIST

10 REM "MSGCREA" NEW 11-22-84 JOHN C. DEBO
20 REM REVISED TO INCLUDE SCREEN EDIT FUNCTIONS 11-23
30 REM MENU CHANGE AND RENUMBER 2-10-85
40 REM
50 REM THIS PROGRAM CREATES AND EDITS THE MESSAGE FILE. THE EMPLOYER'S FILE REFERENCES THESE
60 REM MESSAGES IN ORDER TO GIVE A SPECIAL MESSAGE LIKE "BRING YOUR RESUME" TO THE INTERVIEWER.
70 REM
80 CLS
90 REM
100 REM
110 PRINT " THIS PROGRAM EDITS AND CREATES MESSAGES TO THE INTERVIEWER."
120 PRINT " YOU WILL BE ASKED FOR THE MESSAGE NUMBER FIRST, THEN IF SUCH A"
130 PRINT " MESSAGE EXISTS, IT WILL BE SHOWN TO YOU--IF NO SUCH MESSAGE"
140 PRINT " ALREADY EXISTS, YOU MAY ENTER IT."
150 PRINT " ARE YOU READY TO CONTINUE? (Y/N)"
160 GOSUB 620
170 ON YESNO GOTO 240, 230
180 RUN "A:EXECUTIV.BAS"
190 PRINT
200 PRINT " ENTER THE MESSAGE NUMBER FOLLOWED BY <CR>"
210 INPUT " NUMBER »,N"
220 FILENO = FIX(VAL(N))
230 IF FILENO<1 OR FILENO>99 THEN SOUND 100, 4 : GOTO 250
240 OPEN "R", 1, "B:MESSAGES", 256
250 FIELD 1, 255 AS MESSAGE
260 GET 1, FILENO
270 OUTLINES
280 IF MESSAGE = STRING$(255, 48) THEN 480
290 PRINT " SINCE THERE IS NO MESSAGE NUMBER "; FILENO " DO YOU WISH"
300 PRINT " TO ENTER ONE NOW? (Y/N)"
310 GOSUB 620
320 ON YESNO GOTO 390, 570
330 CLS
340 PRINT " ENTER MESSAGE "; FILENO " OF UP TO 255 CHARACTERS BE LOW"
350 PRINT " "
360 PRINT " "
370 ROW = CR$LIN
380 GOSUB 760
390 LSET MESSAGE = OUTLINES
400 PUT 1, FILENO
410 GOTO 570
420 REM
430 REM THIS SECTION DEALS WITH AN EXISTING MESSAGE.
510 CLS
520 PRINT " MESSAGE NUMBER " #FILENO1 " APPEARS BELOW, YOU MAY EDIT IT AS YOU WI
530 PRINT " PRESS <RETURN> WHEN YOU ARE DONE."
540 PRINT "--------------------------------------------------------------
550 PRINT
560 GOTO 430
570 REM ..............................
580 REM CLOSE THE FILE AND REPEAT
590 REM
600 CLOSE 1
610 GOTO 130
620 REM SUBROUTINE YESNO
630 REM --------------------------------------
640 REM YESNO = 1 FOR YES AND 2 FOR NO ON RETURN.
650 REM
660 REM
670 SOUND 523,3
680 NS = ""
690 WHILE LEN(NS) < 1
700 NS = INKEYS
710 WEND
720 IF NS = "N" OR NS = "n" THEN YESNO = 2 : RETURN
730 IF NS = "Y" OR NS = "y" THEN YESNO = 1 : RETURN
740 SOUND 100,4
750 GOTO 680
760 REM -----------------------------------
770 REM SUBROUTINE INPUT TEXT LINES
780 REM ------------------------------------
790 REM
800 REM ENTER SUBROUTINE WITH "ROW" SHOWING
810 REM WHERE USER WILL START WRITING AND
820 REM "OUTLINE*" TO BE EDITED. IF IT'S
830 REM NEW, OUTLINE* SHOULD BE A DUMMY STRING
840 REM OF 255 BLANKS.
850 REM
860 COL = 1
870 STARTROW = ROW
880 STARTCOL = 1
890 LOCATE ROW, COL
900 PRINT OUTLINE$
910 NS = ""
920 LOCATE ROW, COL
930 A = SCREEN(ROW, COL)
940 WHILE LEN(NS) < 1
950 L = L + 1
960 NS = INKEYS
970 IF ABS(L) = 10 THEN L = 0 - L
980 IF L > 0 THEN CURSOR = 254 ELSE CURSOR = A
990 LOCATE ROW, COL
1000 PRINT CHR$(CURSOR)
1010 WEND
1020 IF LEN(NS) = 2 THEN NS = RIGHT$(NS,1) : GOTO 1140
1030 N = ASC(NS)
1040 IF N = 8 THEN 1140 "TRAP BACKSPACE"
1050 IF N = 13 THEN RETURN
1060 LOCATE ROW, COL
1070 PRINT NS$
1080 CHARNO = (ROW - STARTROW) * 80 + (COL - STARTCOL + 1)
1090 IF CHARNO > 255 THEN RETURN
1100 MID$(OUTLINE$, CHARNO, 1) = NS$
1110 COL = COL + 1
1120 IF COL = 81 THEN COL = 1: ROW = ROW + 1
1130 GOTO 910
1140 REM BACK SPACE, UP, DOWN, RIGHT, AND LEFT ARROWS
1150 REM ALL WORK. CTRL-FUNCTIONS ARE DIFFERENT FROM.
1160 REM WHAT YOU MIGHT EXPECT.
1170 REM
1180 LOCATE ROW,COL
1190 PRINT CHR$(A); 'ERASE CURSOR
1200 N = ASC(N$)
1210 IF N = 77 THEN 1320 'RIGHT ARROW
1220 IF N = 75 THEN 1350 'LEFT ARROW
1230 IF N = 8 THEN 1380 'BACK SPACE
1240 IF N = 72 THEN 1450 'UP ARROW
1250 IF N = 80 THEN 1480 'DOWN ARROW
1260 IF N = 115 THEN 1510 'CTRL-LEFT
1270 IF N = 116 THEN 1570 'CTRL-RIGHT
1280 IF N = 117 THEN 1630 'CTRL-END
1290 IF N = 82 THEN 1730 'INSERT
1300 IF N = 83 THEN 1650 'DELETE
1310 GOTO 910
1320 COL = COL + 1
1330 IF COL = 81 THEN COL = 1:ROW = ROW + 1
1340 GOTO 910
1350 COL = COL - 1
1360 IF COL = 0 THEN COL = 80:ROW = ROW - 1
1370 GOTO 910
1380 LOCATE ROW,COL
1390 PRINT CHR$(32)
1400 CHARNO = (ROW - STARTROW)*80 + (COL - STARTCOL + 1)
1410 MID$(OUTLINE$,CHARNO,1) = " "
1420 COL = COL - 1
1430 IF COL = 0 THEN COL = 80:ROW = ROW - 1
1440 GOTO 910
1450 ROW = ROW - 1
1460 IF ROW < STARTROW THEN ROW = STARTROW
1470 GOTO 910
1480 ROW = ROW + 1
1490 IF ROW > STARTROW + 3 THEN ROW = STARTROW + 3
1500 GOTO 910
1510 COL = COL - 1
1520 COL = COL - 1
1530 IF COL < 1 THEN COL = 1:GOTO 910
1540 IF SCREEN(ROW,COL) <> 32 THEN 1520
1550 COL = COL + 1
1560 GOTO 910
1570 COL = COL + 1
1580 IF COL > 80 THEN COL = 80:GOTO 910
1590 IF SCREEN(ROW,COL) <> 32 THEN 1570
1600 COL = COL + 1
1610 IF COL > 80 THEN COL = 80
1620 GOTO 910
1630 COL = 80
1640 GOTO 910
1650 CHARNO = (ROW - STARTROW)*80 + (COL - STARTCOL + 1)
1660 IF CHARNO > 254 THEN 1700
1670 MID$(OUTLINE$,CHARNO,1) = MID$(OUTLINE$,CHARNO + 1,1)
1680 CHARNO = CHARNO + 1
1690 GOTO 1660
1700 LOCATE STARTROW,STARTCOL
1710 PRINT OUTLINE$
1720 GOTO 910
1730 CHARN0 = (ROW - STARTROW)$80 + (COL - STARTCOL + 1)
1740 P$ = " "
1750 IF CHARN0 > 254 THEN 1700
1760 M$ = MID$(OUTLINE$, CHARN0, 1)
1770 MID$(OUTLINE$, CHARN0, 1) = P$
1780 P$ = M$
1790 CHARN0 = CHARN0 + 1
1800 GOTO 1750
OK
APPENDIX G

MISCELLANEOUS PROGRAMS

These programs are used in looking at files for test purposes, for creating the page drawing files and for start up and access to all the programs previously listed.

FILETEST Functional Description

"FILETEST" lists the contents of any one of the five important random files for trouble shooting purposes. In lines 110-220 a menu is presented which allows the choice of the type of file to be listed. In lines 270 through 320, the file number is ascertained, and then a GOTO statement jumps to the section of code which opens the appropriate type of file and correct file number. Starting at either line 700 or line 940 a format is created for displaying either a large file of up to 750 characters (broken into three pages as the data is used by the programs) or a small file of up to 64 characters. The subroutine "Yesno" is used at the end of these two routines to hold the information on screen until the user is ready to continue. Subroutine "Choice" as usual allows the choice of several options—in this case the five types of random files.
FILETEST Listing

LIST
10 REM
20 REM  FILETEST  11-12-84  JOHN C. DEBO
30 REM  ____________________________________________
40 REM  REVISIONS ON 2-10-85 INCLUDING 'CHOICE'
50 REM
60 REM
70 REM  THIS PROGRAM PRINTS OUT THE CONTENTS OF
80 REM  A RANDOM FILE FOR TROUBLESHOOTING PURPOSES.
90 REM
100 MAXFILE = 100
110 CLS
120 PRINT: PRINT"  FILE TEST": PRINT" WHICH TYPE OF FILE DO YOU WISH TO LOOK AT?"
130 PRINT"  1) THE EMPLOYER FILE"
140 PRINT"  2) THE EMPLOYER CROSS FILE"
150 PRINT"  3) THE STUDENT FILE"
160 PRINT"  4) THE STUDENT CROSS FILE"
170 PRINT"  5) THE EMPLOYER SCHEDULE FILE"
180 PRINT"  WHICH FILE NUMBER SHALL I OPEN?  ENTER 1 TO"
190 PRINT"  "MAXFILE": FOLLOWED BY <CR> ===>";
200 REM
210 REM
220 PRINT: PRINT"  CHOOSE (1 THROUGH 5) PLEASE"
230 CHOICE = 5
240 GOSUB 1200
250 REM
260 REM
270 PRINT:PRINT"  WHICH FILE NUMBER SHALL I OPEN? ENTER 1 TO"
280 PRINT"  "MAXFILE": FOLLOWED BY <CR> ===>";
290 INPUT "", FILENO
300 SOUND 523, 3
310 IF FILENO < 1 OR FILENO > 100 THEN SOUND 100, 4: GOTO 290
320 CLS
330 ON CHOICE GOTO 370, 430, 490, 560, 620
340 REM
350 REM
360 REM
370 OPEN "R", 1, "B:EMPLYRS", 512
380 FIELD 1, 205 AS PG1INFO$, 177 AS PG2INFO$, 130 AS PG3INFO$
390 PRINT "EMPLOYER NUMBER "; FILENO
400 GOTO 680
410 REM
420 REM
430 OPEN "R", 1, "B:EMPLCROS", 64
440 FIELD 1, 64 AS CROSREF$
450 PRINT "EMPLOYER CROSS FILE NUMBER "; FILENO
460 GOTO 940
470 REM
480 REM
490 OPEN "R", 1, "B:STUDENTS", 256
500 FIELD 1, 154 AS PG1INFO$, 102 AS PG2INFO$
510 PG3INFO$ = ""
520 PRINT "STUDENT NUMBER "; FILENO
530 GOTO 680
540 REM
550 REM
560 OPEN "R", 1, "B:STDNCROS", 16
570 FIELD 1, 16 AS CROSREF$  
580 PRINT "STUDENT CROSS FILE NUMBER ";FILENO  
590 GOTO 940  
600 REM  
610 REM  
620 OPEN "R", 1, "B:EMPLSCHE", 512  
630 FIELD 1, 216 AS PG1INFO$, 216 AS PG2INFO$, 80 AS PG3INFO$  
640 PRINT "EMPLOYER SCHEDULE FILE NUMBER ";FILENO  
650 REM  
660 REM  
670 REM  
680 GET 1, FILENO  
690 CLOSE 1  
700 PRINT" |........10........20........30........40........50"  
710 PRINT" | | | | | | | |+ |+ |+ |+ |+ |  
720 PRINT" 1 ""MIDS(PG1INFO$,1,50)  
730 PRINT" 2 ""MIDS(PG1INFO$,101,50)  
740 PRINT" 3 ""MIDS(PG1INFO$,151,50)  
750 PRINT" 4 ""MIDS(PG1INFO$,201,50)  
760 PRINT" 5 ""MIDS(PG1INFO$,251,50)  
770 PRINT" 10 |........20........30........40........50"  
780 PRINT" | | | | | | | |+ |+ |+ |+ |+ |  
790 PRINT" 1 ""MIDS(PG2INFO$,1,50)  
800 PRINT" 2 ""MIDS(PG2INFO$,101,50)  
810 PRINT" 3 ""MIDS(PG2INFO$,151,50)  
820 PRINT" 4 ""MIDS(PG2INFO$,201,50)  
830 PRINT" 5 ""MIDS(PG2INFO$,251,50)  
840 PRINT" |........10........20........30........40........50"  
850 PRINT" | | | | | | | |+ |+ |+ |+ |+ |  
860 PRINT" 1 ""MIDS(PG3INFO$,1,50)  
870 PRINT" 2 ""MIDS(PG3INFO$,101,50)  
880 PRINT" 3 ""MIDS(PG3INFO$,151,50)  
890 PRINT" 4 ""MIDS(PG3INFO$,201,50)  
900 PRINT" 5 ""MIDS(PG3INFO$,251,50)  
910 PRINT"________________>>> CONTINUE? (Y/N) <<<________________

920 GOSUB 1030  
930 ON YESNO GOTO 110, 920  
940 PRINT:PRINT  
950 GET 1, FILENO  
960 CLOSE 1  
970 PRINT" |........10........20........30........40........50........60........70"  
980 PRINT" | | | | | | | |+ |+ |+ |+ |+ |+ |  
990 PRINT " ""CROSREF$  
1000 PRINT:PRINT:PRINT"________________>>> CONTINUE? (Y/N) <<<________________

1010 GOSUB 1030  
1020 ON YESNO GOTO 110, 1010  
1030 REM  
1040 REM SUBROUTINE YESNO  
1050 REM  
1060 REM THIS SUBROUTINE HANDLES YES/NO  
1070 REM QUESTIONS, RETURNING YESNO = 1 FOR  
1080 REM YES AND YESNO = 2 FOR NO  
1090 REM  
1100 SOUND 523, 3  
1110 N$=""  
1120 WHILE LEN(N$) < 1  
1130 N$ = INKEY$
1140 WEND
1150 IF N$ = "N" OR N$ = "n" THEN YESNO = 2:RETURN
1160 IF N$ = "Y" OR N$ = "y" THEN YESNO = 1:RETURN
1170 SOUND 100,4
1180 GOTO 1110
1190 REM -----------------------------
1200 REM SUBROUTINE CHOICE
1210 REM ---------------------------------
1220 REM THIS SUBROUTINE HANDLES CHOICES OF
1230 REM MORE THAN ONE OPTION. CALL IT WITH
1240 REM CHOICE = NUMBER OF LARGEST OPTION; IT
1250 REM RETURNS WITH CHOICE = NUMBER OF OPTION
1260 REM CHOSEN.
1270 REM
1280 SOUND 523,3
1290 N$ = ""
1300 WHILE LEN(N$) < 1
1310 N$ = INKEY$;
1320 WEND
1330 N = FIX(VAL(N$))
1340 IF N < 1 OR N > CHOICE THEN SOUND 100,4: GOTO 1290
1350 CHOICE = N
1360 RETURN
Ok
CREAPG Functional Description

CREAPG1, CREAPG2, CREAPG3, and CREAPG4 work in the same way. The first two data statements are used to load printed characters and words into the CHAR$ array. These are used to make up the forms which EMPLUP and STDNUP put on the screen. The next data statement gives the row, column locations, the number of characters to be read and the column numbers to skip while reading data from the form. These are stored in the array LOCAT%. The final data statement (DRW%) contains the information used in drawing the form. It places the characters and words from the first two data statements on the form. It contains the row and column locations, the number of times to repeat that character or word, and the character number (out of array CHAR$).

The remaining task for the programs is to write CHAR$, LOCAT%, and DRW$ into the file and close it. These programs only need to be run if some change is made to them in order to make some change in the forms.
CREAPGl Listing

```
LIST
10 REM -----------------------------------------------
20 REM CREAPGl 7-7-84 JOHN C. DEBO 
30 REM -----------------------------------------------
40 REM REVISED 8-10-84 ADDITIONAL COMMENTS 
50 REM RENUMBERED 2-10-85
60 REM 
80 REM CREAPGl IS USED TO CREATE THE SEQUENTIAL 
90 REM FILE (PAGEONE) USED IN DRAWINT THE 'FORM' 
100 REM USED IN INPUT/OUTPUT OF DATA IN STDMUP.
110 REM
120 REM ARRAY CHAR* HOLDS GRAPHICS CHARACTERS 
130 REM AND WORDS OF THE FORM
140 REM 
150 REM ARRAY LOCATX HOLDS LOCATION INFORMATION 
160 REM FOR PICKING UP DATA FROM THE SCREEN
170 REM
180 REM ARRAY DRWX HOLDS LOCATIONS AND NUMBER 
190 REM OF CHARACTERS REQUIRED TO DRAW THE FORM 
200 REM USING CHARACTERS FROM CHAR*
210 REM 
220 REM OPTION BASE 1 'ARRAY SUBSCRIPTS START WITH 1 (NOT 0)
230 DIM CHAR*(60) '* INDICATES ASCII - 1 BYTE 
240 DIM LOCATX(150) '% INDICATES INTEGER - 2 BYTE
250 DIM DRWX(400) 
260 REM-------------------------------------------------------
270 REM THESE ARRAY SIZES MAY BE INCREASED--YOU ALSO WOULD 
280 REM NEED TO CHANGE ARRAY SIZES IN USING PROGRAMS. NOTE 
290 REM THAT THE USING PROGRAMS EMPLOY STOPPERS (# AND -1) TO 
300 REM TELL THEM WHERE THE LAST CHARACTERS ARE IN THE ARRAYS.
310 REM--------------------------------------------------------
320 REM CREATE CHAR* -- FIRST THE GRAPHICS CHARACTERS 
330 REM 
340 REM P1 IS JUST A COUNTER, X IS THE DECIMAL 
350 REM 
360 REM CHAR*(P1)=CHR*(X)
370 REM NEXT P1
380 DATA 196,179,192,193,217,218,194,191,195,197,180,205,212,207,190,219,219, 
390 REM 
400 REM CHRS IS THE BASIC FUNCTION WHICH CONVERTS THE 
410 REM DECIMAL NUMBER TO A GRAPHICS CHARACTER.
420 FOR P1 = 1 TO 20 
430 READ X 
440 CHAR*(P1)=CHR*(X)
450 NEXT P1
460 REM CREATING CHAR* -- ADD IN THE WORDS NOW 
470 REM READ CHAR*(P1)
480 IF CHAR*(P1) = "#" THEN 650 'THIS IS THE "STOPPER"
490 P1=P1+1
500 GOTO 470
```
DATA STUDENT, (LAST), (FIRST), (MIDDLE), SOCIAL SECURITY, NAME, INITIAL), NUMBER, TELEPHONE, (AT HOME), (ANOTHER), -- THE SECOND IS: AT WORK = 1, NUMBER, OR AT PARENTS = 2
DATA OR AT FRIENDS = 3, LOCAL, (STREET), (CITY), (STATE), (ZIP), ADDRESS, PERMANENT, (COUNTRY), *
DATA OR AT FRIENDS = 4, LOCAL, (STREET), (CITY), (STATE), (ZIP), ADDRESS, PERMANENT, (COUNTRY), *

530 REM
540 REM CREATE LOCATX -- THE DATA LOCATION FILE
550 REM
560 REM -----------------------------------------------
570 REM DATA FORMAT -- ROW, COL, NCHAR, NSKIP1, NSKIP2
580 REM
590 REM ROW & COL GIVE LOCATION OF 1ST CHARACTER
600 REM
610 REM NCHAR GIVES NUMBER OF CHARACTERS TO READ
620 REM
630 REM NSKIP 1 & NSKIP 2 GIVE COLUMN NUMBERS TO
640 REM SKIP AS REQUIRED
650 REM
660 P1 = 1
670 READ LOCATX(P1)
680 IF LOCATX(P1) = -1 THEN B400 'THIS IS THE "STOPPER"
690 P1 = P1 + 1
700 GOTO 670
710 DATA 8, 12, 15, 0, 0, 8, 30, 14, 0, 0, 8, 49, 1, 0, 0, 8, 61, 9, 64, 67, 13, 14, 10, 17, 21, 13, 31, 10, 34, 38, 13, 49, 1, 0, 0, 18, 13, 20, 0, 0, 18, 36, 15, 0, 0
720 DATA 23, 54, 2, 0, 0, 23, 59, 5, 0, 0, 23, 67, 10, 0, 0, 0, 0, 0, 0, -1
730 REM
740 REM CREATE DRWX -- THE PAGE DRAWING FILE
750 REM
760 REM -----------------------------------------------
770 REM DATA FORMAT -- ROW, COL, NPRINT, CHAR
780 REM
790 REM NPRINT GIVES THE NUMBER OF REPETIONS OF A CHARACTER
800 REM
810 REM CHAR IS THE LOCATION IN CHAR* OF THE GRAPHIC
820 REM CHARACTER OR WORD TO BE DRAWN ON THE PAGE
830 REM
840 P1 = 1
850 READ DRWX(P1)
860 IF DRWX(P1) = -1 THEN 1050 'THIS IS THE "STOPPER"
870 P1 = P1 + 1
880 GOTO 850
890 DATA 5, 1, 79, 12, 6, 3, 1, 21, 6, 17, 1, 22, 6, 34, 1, 23, 6, 46, 1, 24, 6, 59, 1, 25, 7, 3, 1, 26, 7, 46, 1, 27, 7, 63, 1, 28, 9, 11, 1, 2, 9, 12, 15, 1, 9, 27, 1, 5, 9, 29, 1, 3, 9, 30, 14, 1
900 DATA 9, 44, 1, 5, 9, 48, 1, 3, 9, 49, 1, 1, 9, 50, 1, 5, 9, 60, 1, 3, 9, 61, 3, 1, 9, 64, 1, 4, 9, 65, 2, 1, 9, 67, 1, 4, 9, 68, 4, 1, 9, 72, 1, 5, 11, 3, 1, 29, 11, 16, 1, 30, 11, 33, 1, 31, 11, 45, 1, 32
910 DATA 12, 3, 1, 33, 12, 53, 1, 34, 14, 13, 1, 3, 14, 14, 3, 1, 14, 17, 1, 4, 14, 18, 3, 1, 14, 21, 1, 4, 14, 22, 4, 1, 14, 26, 1, 5, 14, 30, 1, 3, 14, 31, 3, 1, 14, 34, 1, 4, 14, 35, 3, 1, 14, 38, 1, 4
920 DATA 14, 39, 4, 1, 14, 43, 1, 5, 14, 48, 1, 3, 14, 49, 1, 1, 14, 50, 1, 5, 14, 53, 1, 3, 14, 54, 1, 3, 14, 56, 1, 5, 14, 58, 1, 3, 14, 59, 5, 1, 17, 3, 1, 41
930 DATA 19, 12, 1, 3, 19, 13, 20, 1, 19, 33, 1, 5, 19, 35, 1, 3, 19, 36, 1, 5, 19, 51, 1, 5, 19, 53, 1, 3, 19, 54, 2, 1, 19, 56, 1, 5, 19, 58, 1, 3, 19, 59, 5, 1,
19, 64, 1, 5, 21, 3, 1, 42
940 DATA 21, 19, 1, 37, 21, 41, 1, 38, 21, 51, 1, 39, 21, 59, 1, 40, 21, 67, 1, 43, 22, 3, 1, 41, 24, 12, 1, 3, 24, 13, 20, 1, 24, 33, 1, 5, 24, 35, 1, 3, 24, 36, 1, 5, 24, 51, 1, 5
950 DATA 24, 53, 1, 3, 24, 54, 2, 1, 24, 56, 1, 5, 24, 58, 1, 3, 24, 59, 5, 1, 24, 64, 1, 5, 24, 66, 1, 3, 24, 67, 10, 1, 24, 77, 1, 5, 0, 0, 0, 0, -1
CREATE "PAGEONE", ETC -- THE FILE WHICH
ALLOWS USING A "PAGE" FOR OPERATOR INPUT OR
OUTPUT TO THE OPERATOR OF DATA.

SINCE "CREAPG" IS USED AS THE NAME OF SEVERAL
PROGRAMS--BE CAREFUL OF THE PAGE NUMBER!!

OPEN "O",1,"A:PAGEONE"

P1=1
WRITE #1,CHARS(P1)
IF CHAR$(P1) = "#" THEN 1110
P1 = P1 + 1
GOTO 1070

P1 = 1
WRITE #1,LOCATY(P1)
IF LOCATY(P1) = -1 THEN 1160
P1 = P1 + 1
GOTO 1120

P1 = 1
WRITE #1,DRWX(P1)
IF DRWX(P1) = -1 THEN 1210
P1 = P1 + 1
GOTO 1170
CLOSE 1
END
CREAPG2 Listing

LIST
10 REM
20 REM CREAPG2 8-15-84 JOHN C. DEBO
30 REM
40 REM RENUMBERED 2-10-85
50 REM
60 REM CREAPG2 IS USED TO CREATE THE SEQUENTIAL
70 REM FILE (PAGETWO) USED IN DRAWING THE 'FORM'
80 REM FOR INPUT AND OUTPUT OF DATA IN STTNUP.
90 REM
100 REM
110 REM
120 REM ARRAY CHAR$ HOLDS GRAPHICS CHARACTERS
130 REM AND WORDS OF THE FORM
140 REM
150 REM ARRAY LOCATX HOLDS LOCATION INFORMATION
160 REM FOR PICKING UP DATA FROM THE SCREEN
170 REM
180 REM ARRAY DRW$ HOLDS LOCATIONS AND NUMBER
190 REM OF CHARACTERS REQUIRED TO DRAW THE FORM
200 REM USING CHARACTERS FROM CHAR$
210 REM
220 OPTION BASE 1 'ARRAY SUBSCRIPTS START WITH 1 (NOT 0)
230 DIM CHAR$(60) '* INDICATES ASCII - 1 BYTE
240 DIM LOCATX(150) '* INDICATES INTEGER - 2 BYTE
250 DIM DRW$(450)
260 REM
270 REM THESES ARRAY SIZES MAY BE INCREASED--YOU ALSO WOULD
280 REM NEED TO CHANGE ARRAY SIZES IN USING PROGRAMS. NOTE
290 REM THAT THE USING PROGRAMS EMPLOY STOPPERS (# AND -1) TO
300 REM TELL THEM WHERE THE LAST CHARACTERS ARE IN THE ARRAYS.
310 REM
320 REM CREATE CHAR$ -- FIRST THE GRAPHICS CHARACTERS
330 REM
340 REM P1 IS JUST A COUNTER, X IS THE DECIMAL
350 REM VALUE OF A GRAPHICS CHARACTER
360 REM
370 REM CHAR$ IS THE BASIC FUNCTION WHICH CONVERTS THE
380 REM DECIMAL NUMBER TO A GRAPHICS CHARACTER.
390 REM
400 FOR P1 = 1 TO 20
410 READ X
420 CHAR$(P1) = CHR$(X)
430 NEXT P1
440 DATA 196,179,192,193,217,218,194,191,195,197,180,205,212,207,190,219,219,
219,219,219
450 REM CREATING CHAR$ -- ADD IN THE WORDS NOW
460 READ CHAR$(P1)
470 IF CHAR$(P1) = "#" THEN 640 'THIS IS THE "STOPPER"
480 P1 = P1 + 1
490 GOTO 460
500 DATA GRAD DATE--TERM YEAR, MAJOR CODE, INTERVIEW HISTORY-, 1) FALL, 2) SPR, 3)
SUM, (MO.DA.YR) FILE NO., CITIZENSHIP--CODE DEGREE--CODE, 1) U.S.A., 2) BACH., 2)
PERM VISA, 2) MASTER, 3) WORK PRMT, 3) DOCTOR, 4) CERTF.
510 DATA (MO.DA.YR) FILE NO.,GRADE POINT--AS OF, (X.XX) TERM YEAR,*
520 REM
530 REM CREATE LOCATX -- THE DATA LOCATION FILE
540 REM --------------------------------------------
550 REM DATA FORMAT -- ROW, COL, NCHAR, NSKIP1, NSKIP2
560 REM ROW & COL GIVE LOCATION OF 1ST CHARACTER
570 REM NCHAR GIVES NUMBER OF CHARACTERS TO READ
580 REM NSKIP 1 & NSKIP 2 GIVE COLUMN NUMBERS TO READ
590 REM SKIP AS REQUIRED
600 REM
610 REM READ LOCATX(P1)
620 IF LOCATX(P1) = -1 THEN 830 'THIS IS THE "STOPPER"
630 P1 = P1 + 1
640 DATA 7,15,1,0,0, 7,21,2,0,0, 7,30,4,0,0, 13,18,1,0,0, 13,33,1,0,0,
650 20,7,3,8,0, 20,17,1,0,0, 20,21,2,0,0, 7,47,6,49,52, 7,60,4,0,0
660 DATA 11,47,6,49,52, 11,60,4,0,0, 13,47,6,49,52, 13,60,4,0,0,
670 18,47,6,49,52, 18,60,4,0,0, 20,47,6,49,52, 20,60,4,0,0,
680 23,47,6,49,52, 23,60,4,0,0, 0,0,0,0,-1
690 DATA 7,15,1,0,0, 7,21,2,0,0, 7,30,4,0,0, 13,18,1,0,0, 13,33,1,0,0,
700 20,7,3,8,0, 20,17,1,0,0, 20,21,2,0,0, 7,47,6,49,52, 7,60,4,0,0
710 DATA 11,47,6,49,52, 11,60,4,0,0, 13,47,6,49,52, 13,60,4,0,0,
720 18,47,6,49,52, 18,60,4,0,0, 20,47,6,49,52, 20,60,4,0,0,
730 23,47,6,49,52, 23,60,4,0,0, 0,0,0,0,-1
740 DATA
750 REM CREATE DRWX -- THE PAGE DRAWING FILE
760 REM DATA FORMAT -- ROW, COL, NPRINT, CHAR
770 REM NPRINT GIVES THE NUMBER OF REPETITIONS OF A CHARACTER
780 REM CHAR IS THE LOCATION IN CHAR* OF THE GRAPHIC CHARACTER OR WORD TO BE DRAWN ON THE PAGE
790 REM
800 REM READ DRWX(P1)
810 IF DRWX(P1) = -1 THEN 1030 'THIS IS THE "STOPPER"
820 P1 = P1 + 1
830 DATA 5,1,79,12, 6,3,1,21, 6,27,1,22, 6,46,1,23, 8,3,1,24, 8,14,1,3,
840 8,15,1,1, 8,16,1,5, 8,20,1,3, 8,21,2,1, 8,23,1,5, 8,29,1,3,
850 8,30,4,1, 8,34,1,5, 8,46,1,3, 8,47,2,1, 8,49,1,4, 8,50,2,1
860 DATA 9,3,1,25, 10,3,1,26, 10,46,1,27, 12,3,1,28, 12,46,1,3, 12,47,2,1,
870 12,49,1,4, 12,50,2,1, 12,52,1,4, 12,53,2,1, 12,55,1,5, 12,59,1,3
880 DATA 12,60,4,1, 12,64,1,5, 14,3,1,29, 14,17,1,3, 14,18,1,1, 14,19,1,5,
890 14,22,1,30, 14,32,1,3, 14,33,1,1, 14,34,1,5, 14,41,1,4, 14,47,2,1,
900 14,49,1,4, 14,50,2,1, 14,52,1,4, 14,53,2,1, 14,55,1,5, 14,59,1,3
910 DATA 14,60,4,1, 14,64,1,5, 15,3,1,31, 15,22,1,32, 16,3,1,33, 16,22,1,34,
920 17,22,1,35, 17,46,1,36, 19,3,1,37, 19,46,1,3, 19,47,2,1, 19,49,1,4,
930 19,50,2,1, 19,52,1,4, 19,53,2,1, 19,55,1,5, 19,59,1,3, 19,60,4,1
940 DATA 19,64,1,5, 21,6,1,3, 21,7,1,1, 21,8,1,4, 21,9,2,1, 21,11,1,5,
950 21,16,1,3, 21,17,1,1, 21,18,1,5, 21,20,1,3, 21,21,2,1, 21,23,1,5,
960 21,46,1,3, 21,47,2,1, 21,49,1,4, 21,50,2,1, 21,52,1,4, 21,53,2,1
970 DATA 21,55,1,5, 21,59,1,3, 21,60,4,1, 21,64,1,5, 24,46,1,3, 24,47,2,1,
980 24,49,1,4, 24,50,2,1, 24,52,1,4, 24,53,2,1, 24,55,1,5, 24,59,1,3,
990 24,60,4,1, 24,64,1,5, 22,6,1,38, 0,0,0,0,-1
990 REM SINCE "CREAPG" IS USED AS THE NAME OF SEVERAL
1000 REM PROGRAMS--BE CAREFUL OF THE PAGE NUMBER!!
1020 REM
1030 OPEN "O",1,"A:PAGE TWO"
1040 P1 = 1
1050 WRITE #1,CHAR*(P1)
1060 IF CHAR*(P1) = "2" THEN 1090
1070 P1 = P1 + 1
1080 GOTO 1050
1090 P1 = 1
1100 WRITE #1,LOCATX(P1)
1110 IF LOCATX(P1) = -1 THEN 1140
1120 P1 = P1 + 1
1130 GOTO 1100
1140 P1 = 1
1150 WRITE #1,DRWX(P1)
1160 IF DRWX(P1) = -1 THEN 1190
1170 P1 = P1 + 1
1180 GOTO 1150
1190 CLOSE 1
1200 END
0k
CREAPG3 Listing

LIST
10 REM
20 REM CREAPG3 9-8-84 JOHN C. DEBO
30 REM
40 REM RENUMBERED 2-10-85
50 REM
60 REM
70 REM CREAPG3 IS USED TO CREATE THE SEQUENTIAL
80 REM FILE (PAGETHRE) USED IN MAKING THE 'FORM'
90 REM USED TO INPUT/OUTPUT DATA IN EMPLUP.
100 REM
110 REM
120 REM ARRAY CHAR$ HOldS GRAPHiCS CHARACTERS
130 REM AND WORDS OF THE FORM
140 REM
150 REM ARRAY LOCATX HOldS LOCATION INFORMATION
160 REM FOR PICKING UP DATA FROM THE SCREEN
170 REM
180 REM ARRAY DRWX HOldS LOCATIONS AND NUMBER
190 REM OF CHARACTERS REQUIRED TO DRAW THE FORM
200 REM USING CHARACTERS FROM CHAR$
210 REM
220 OPTION BASE 1 ' ARRAY SUBSCRIPTS START WITH 1 (NOT 0)
230 DIM CHAR$(60) ' * INDICATES ASCII - 1 BYTE
240 DIM LOCATX(250) ' * INDICATES INTEGER - 2 BYTE
250 DIM DRWX(800)
260 REM
270 REM THESE ARRAY SIZES MAY BE INCREASED--YOU ALSO WOULD
280 REM NEED TO CHANGE ARRAY SIZES IN USING PROGRAMS. NOTE
290 REM THAT THE USING PROGRAMS EMPLOY STOPPERS (# AND -1) TO
300 REM TELL THEM WHERE THE LAST CHARACTERS ARE IN THE ARRAYS.
310 REM
320 REM CREATE CHAR$ -- FIRST THE GRAPHiCS CHARACTERS
330 REM
340 REM P1 IS JUST A COUNTER, X IS THE DECIMAL
350 REM VALUE OF A GRAPHICS CHARACTER
360 REM
370 REM CHR$ IS THE BASIC FUNCTION WHICH CONVERTS THE
380 REM DECIMAL NUMBER TO A GRAPHICS CHARACTER.
390 REM
400 FOR P1 = 1 TO 20
410 READ X
420 CHAR$(P1) = CHR$(X)
430 NEXT P1
440 DATA 196,179,192,193,217,218,194,191,195,197,180,205,212,207,190,219,219,
450 DATA 219,219,219
460 REM CREATING CHAR$ -- ADD IN THE WORDS NOW
470 IF CHAR$(P1) = "#" THEN 640 ' THIS IS THE "STOPPER"
480 P1 = P1 + 1
490 GOTO 460
500 DATA COMPANY NAME, "CONTACT: (LAST) (FIRST) (M/I) "DIVISION,TITLE
510 , ADDRESS (STREET), PHONE, EXTENSION, (CITY),(STATE), "MESSAGES TO INTERVIEWEES: ",
520 ZIP CODE (COUNTRY), GRADUATION DATES (TERM/YEAR)
530 DATA "1=FALL, 2=SPRG, 3=SUMR, 4=ALL", "
540 REM
550 REM CREATE LOCATX -- THE DATA LOCATION FILE
560 REM DATA FORMAT -- ROW, COL, NCHAR, NSkip1, NSkip2
REM ROW & COL GIVE LOCATION OF 1ST CHARACTER
REM NCHAR GIVES NUMBER OF CHARACTERS TO READ
REM NSKIP 1 & NSKIP 2 GIVE COLUMN NUMBERS TO
REM SKIP AS REQUIRED

PI = 1

READ LOCATX(P1)

IF LOCATX(P1) = -1 THEN 840  'THIS IS THE "STOPPER"

PI = PI + 1

GOTO 660

DATA 6, 6, 25, 0, 0, 10, 6, 25, 0, 0, 14, 6, 25, 0, 0, 18, 6, 15, 0, 0, 18, 25, 2, 0, 0,
22, 6, 9, 11, 0, 22, 20, 10, 0, 0

DATA 6, 41, 15, 0, 0, 6, 58, 14, 0, 0, 6, 74, 1, 0, 0, 10, 41, 25, 0, 0, 14, 41, 10, 44, 48,
14, 58, 7, 61, 0, 18, 41, 2, 0, 0, 18, 47, 2, 0, 0, 18, 53, 2, 0, 0, 18, 59, 2, 0, 0,
18, 65, 2, 0, 0, 22, 41, 3, 42, 0, 22, 48, 3, 49, 0, 22, 55, 3, 56, 0

DATA 22, 62, 3, 63, 0, 0, 0, 0, 0, 0, -1

CREATE DRWX -- THE PAGE DRAWING FILE

DATA FORMAT -- ROW,COL,NPRINT,CHAR

NPRINT GIVES THE NUMBER OF REPETITIONS OF A CHARACTER
CHAR IS THE LOCATION IN CHAR* OF THE GRAPHIC CHARACTER OR WORD TO BE DRAWN ON THE PAGE

READ DRWX<P1>

IF DRWX<P1> = -1 THEN 1040  'THIS IS THE "STOPPER"

PI = PI + 1

GOTO 850

DATA 4, 1, 79, 12, 5, 3, 1, 21, 5, 36, 1, 22, 7, 5, 1, 3, 7, 6, 25, 1, 7, 31, 1, 5,
7, 40, 1, 3, 7, 41, 15, 1, 7, 56, 1, 5, 7, 57, 1, 3, 7, 58, 1, 4, 7, 72, 1, 5,
7, 73, 1, 3, 7, 74, 1, 1, 7, 75, 1, 5, 9, 3, 1, 23, 9, 38, 1, 24

DATA 11, 5, 1, 3, 11, 6, 25, 1, 11, 31, 1, 5, 11, 40, 1, 3, 11, 41, 25, 1, 11, 66, 1, 5,
13, 3, 1, 25, 13, 38, 1, 26, 13, 57, 1, 27, 15, 5, 1, 3, 15, 6, 25, 1, 15, 31, 1, 5,
15, 40, 1, 3, 15, 41, 3, 1, 15, 44, 1, 4, 15, 45, 3, 1, 15, 48, 1, 4, 15, 49, 4, 1

DATA 15, 53, 1, 5, 15, 57, 1, 3, 15, 58, 3, 1, 15, 61, 1, 4, 15, 62, 4, 1, 15, 66, 1, 5,
17, 3, 1, 28, 17, 23, 1, 29, 17, 38, 1, 30, 19, 5, 1, 3, 19, 6, 1, 5, 19, 6, 15, 1, 19, 21, 1, 5,
19, 24, 1, 3, 19, 25, 2, 1, 19, 27, 1, 5, 19, 40, 1, 3, 19, 41, 2, 1, 19, 43, 1, 5

DATA 19, 46, 1, 3, 19, 47, 2, 1, 19, 49, 1, 5, 19, 52, 1, 3, 19, 53, 2, 1, 19, 55, 1, 5,
19, 58, 1, 3, 19, 59, 2, 1, 19, 61, 1, 5, 19, 64, 1, 4, 19, 65, 2, 1, 19, 67, 1, 5,
21, 3, 1, 31, 21, 38, 1, 32, 23, 5, 1, 3, 23, 6, 5, 1, 23, 11, 1, 4, 23, 12, 4, 1

DATA 23, 16, 1, 5, 23, 19, 1, 3, 23, 20, 1, 10, 1, 23, 30, 1, 5, 23, 40, 1, 3, 23, 41, 1, 1,
23, 42, 1, 4, 23, 43, 2, 1, 23, 45, 1, 5, 23, 47, 1, 3, 23, 48, 1, 1, 23, 49, 1, 4,
23, 50, 2, 1, 23, 52, 1, 5, 23, 54, 1, 3, 23, 55, 1, 1, 23, 56, 1, 4, 23, 57, 2, 1

DATA 23, 59, 1, 5, 23, 61, 1, 3, 23, 62, 1, 1, 23, 63, 1, 4, 23, 64, 2, 1, 23, 66, 1, 5,
24, 38, 1, 33, 0, 0, 0, -1

CREATE "PAGETHRE", ETC -- THE FILES WHICH
ALLOW USING A "PAGE" FOR OPERATOR INPUT OR
OUTPUT OF DATA.

SINCE "CREAPG" IS USED AS THE NAME OF SEVERAL
PROGRAMS--BE CAREFUL OF THE PAGE NUMBER!!
1040 OPEN "O",1,"A\:PAGETHRE"
1050 P1=1
1060 WRITE #1,CHAR*(P1)
1070 IF CHAR*(P1) = "I" THEN 1100
1080 P1 = P1 + 1
1090 GOTO 1060
1100 P1 = 1
1110 WRITE #1,LOCAT%(P1)
1120 IF LOCAT%(P1) = -1 THEN 1150
1130 P1 = P1 + 1
1140 GOTO 1110
1150 P1 = 1
1160 WRITE #1,DRW%(P1)
1170 IF DRW%(P1) = -1 THEN 1200
1180 P1 = P1 + 1
1190 GOTO 1160
1200 CLOSE 1
1210 END
0k
CREAPG4 Listing

LIST
10 REM
20 REM CREAPG4 9-2-84 JOHN C. DEBO
30 REM
40 REM REVISEd 8-10-84 ADDITIONAL COMMENTS
50 REM REVISEd 11-3-84 ADD G.P.A. AND 2 HRS
60 REM RENUMBERED 2-10-85
70 REM
80 REM CREAPG4 IS USED TO CREATE THE SEQUENTIAL
90 REM FILE (PAGEONE) USED IN DRAWING THE 'FORM'
100 REM USED IN INPUT/OUTPUT WORK IN ENMLUP.
110 REM
120 REM
130 REM
140 REM ARRAY CHAR$ HOLDS GRAPHICS CHARACTERS
150 REM AND WORDS OF THE FORM
160 REM
170 REM ARRAY LOCAT$ HOLDS LOCATION INFORMATION
180 REM FOR PICKING UP DATA FROM THE SCREEN
190 REM
200 REM ARRAY DRW$ HOLDS LOCATIONS AND NUMBER
210 REM OF CHARACTERS REQUIRED TO DRAW THE FORM
220 REM USING CHARACTERS FROM CHAR$
230 REM
240 OPTION BASE 1 'ARRAY SUBSCRIPTS START WITH 1 (NOT 0)
250 DIM CHAR$(60) 'S INDICATES ASCII - 1 BYTE
260 DIM LOCAT$(250) '% INDICATES INTEGER - 2 BYTE
270 DIM DRW$(800)
280 REM --------------------------------------------------------
290 REM THESE ARRAY SIZES MAY BE INCREASED--YOU ALSO WOULD
300 REM NEED TO CHANGE ARRAY SIZES IN USING PROGRAMS. NOTE
310 REM THAT THE USING PROGRAMS EMPLOY STOPPERS (%) AND -1) TO
320 REM TELL THEM WHERE THE LAST CHARACTERS ARE IN THE ARRAYS.
330 REM --------------------------------------------------------
340 REM CREATE CHAR$ -- FIRST THE GRAPHICS CHARACTERS
350 REM
360 REM P1 IS JUST A COUNTER, X IS THE DECIMAL
370 REM VALUE OF A GRAPHICS CHARACTER
380 REM
390 REM CHR$ IS THE BASIC FUNCTION WHICH CONVERTS THE
400 REM DECIMAL NUMBER TO A GRAPHICS CHARACTER.
410 REM
420 FOR P1 = 1 TO 20
430 READ X
440 CHAR$(P1)=CHR$(X)
450 NEXT P1
460 DATA 196,179,192,193,217,218,194,191,195,197,180,205,212,207,190,219,219,
219,219,219
470 REM CREATING CHAR$ -- ADD IN THE WORDS NOW
480 READ CHAR$(P1)
490 IF CHAR$(P1) = "%" THEN 660 'THIS IS THE "STOPPER"
500 P1=P1+1
510 GOTO 480
520 DATA MAJOR CODES TO BE INTERVIEWED, INTVWRS DATE NO.OF INTVWS,"(ALL M
AJ.= 000, ALL ENGR.= 0900, ETC)",(NUMBER) (MD/DA/YR) AM PM,DEGREES,CITIZENSHIP
,1) BACH. 3) DOCTOR 1) Req. U.S.A.
**DATA 2) MASTER 4) CERTF.** 2) PERM VISA, 3) WORK PERMIT, WORK LOCATION(S), LENGTH OF INTERVIEW, "(1=30, 2=45, 3=60 MIN)", "POSITION NAME(S):", START TIMES AM PM, (HR/MIN), "MINIMUM GPA: ";

**540 REM CREATE LOCAT% -- THE DATA LOCATION FILE**

**560 REM**

**580 REM DATA FORMAT -- ROW, COL, NCHAR, NSKIP1, NSKIP2**

**600 REM**

**620 REM NCHAR & COL GIVE LOCATION OF 1ST CHARACTER**

**640 REM**

**660 REM**

**670 Pl = 1**

**680 READ LOCAT%(P1)**

**690 IF LOCAT%(P1) = -1 THEN B60 'THIS IS THE "STOPPER"**

**700 Pl = Pl + 1**

**710 GOTO 680**

**720 DATA 6,6,4,0,0,6,13,4,0,0,6,20,4,0,0,6,27,4,0,0,6,34,4,0,0,**

**8,6,4,0,0,8,13,4,0,0,8,20,4,0,0,8,27,4,0,0,8,34,4,0,0,**

**13,6,1,0,0,13,10,1,0,0,13,14,1,0,0,13,18,1,0,0,13,32,1,0,0**

**730 DATA 19,4,25,0,0,21,21,17,0,0,23,5,33,0,0,6,47,1,0,0,**

**6,54,6,56,59,8,67,1,0,0,8,70,1,0,0,11,47,1,0,0,11,54,6,56,59,**

**11,67,1,0,0**

**740 DATA 15,70,1,0,0,15,47,1,0,0,15,54,6,56,59,15,67,1,0,0,**

**15,70,1,0,0,23,56,4,58,0,23,64,4,66,0,0,0,0,-1**

**750 REM**

**760 REM CREATE DRW% -- THE PAGE DRAWING FILE**

**770 REM**

**780 REM**

**790 REM DATA FORMAT -- ROW, COL, NPRINT, CHAR**

**800 REM**

**810 REM NPRINT GIVES THE NUMBER OF REPETITIONS OF A CHARACTER**

**820 REM**

**830 REM**

**840 REM**

**850 REM**

**860 Pl = 1**

**870 READ DRW%(P1)**

**880 IF DRW%(P1) = -1 THEN 1100 'THIS IS THE "STOPPER"**

**890 Pl = Pl + 1**

**900 GOTO 870**

**910 DATA 4,1,79,12,5,3,1,21,5,44,1,22,7,5,1,3,7,6,4,1,**

**7,10,1,5,7,12,1,3,7,13,4,1,7,17,1,5,7,19,1,3,**

**7,20,4,1,7,24,1,5,7,26,1,3,7,27,4,1,7,31,1,5,**

**7,33,1,3,7,34,4,1,7,38,1,5,7,46,1,3,7,47,1,1,**

**7,48,1,5,7,53,1,3,7,54,2,1,7,56,1,4,7,57,2,1,**

**7,59,1,4,7,60,2,1,7,62,1,5,7,66,1,3,7,67,1,1,**

**7,68,1,5,7,69,1,3,7,70,1,1,7,71,1,5,9,5,1,3,**

**9,6,4,1,9,10,1,5,7,12,1,3,7,13,4,1,7,17,1,5,**

**7,19,1,3,7,20,4,1,7,24,1,5,7,26,1,3,7,27,4,1,**

**7,31,1,5,7,33,1,3,7,34,4,1,7,38,1,5,7,46,1,3,**

**7,47,1,1,7,48,1,5,7,53,1,3,7,54,2,1,7,56,1,4,**

**7,57,2,1,7,59,1,4,7,60,2,1,7,62,1,5,7,66,1,3,**

**7,67,1,1,7,68,1,5,7,69,1,3,7,70,1,1,7,71,1,5,**

**9,5,1,3,9,6,4,1,9,9,1,3**

**930 DATA 9,10,1,5,9,12,1,3,9,13,4,1,9,17,1,5,9,19,1,3,**

**9,20,4,1,9,24,1,5,9,26,1,3,9,27,4,1,9,31,1,5,**

**9,33,1,3,9,34,4,1,9,36,1,5,9,46,1,3,9,47,1,1,**

**9,48,1,5,9,53,1,3,9,54,2,1,9,56,1,4,9,57,2,1,**

**9,59,1,4,9,60,2,1,9,62,1,5,9,66,1,3,9,67,1,1,**

**9,68,1,5,9,69,1,3,9,71,1,1,9,71,1,5,10,3,1,23,**

**10,44,1,24,12,3,1,25,12,24,1,26,12,46,1,3,**

**12,47,1,1,12,48,1,5,12,53,1,3,12,54,2,1,**

**12,56,1,4,12,57,2,1,12,59,1,4,12,60,2,1,**

**12,62,1,5,12,66,1,3,12,67,1,1,12,68,1,5,12,69,1,3,**

**12,70,1,1,12,71,1,5,14,5,1,3,14,6,1,1,**

**14,7,1,5,14,9,1,3,14,10,1,1**
960 DATA 14,11,1,5, 14,13,1,3, 14,14,1,1, 14,15,1,5, 14,17,1,3, 14,18,1,1, 14,19,1,5, 14,31,1,3, 14,32,1,1, 14,33,1,5, 14,46,1,3, 14,47,1,1, 14,48,1,5, 14,53,1,3, 14,54,2,1, 14,56,1,4, 14,57,2,1, 14,59,1,4, 14,60,2,1, 14,62,1,5, 14,66,1,3, 14,67,1,1, 14,68,1,5, 14,69,1,5, 14,70,1,1, 14,71,1,5, 15,3,1,27, 16,3,1,28, 16,46,1,3, 16,47,1,1, 16,48,1,5, 16,53,1,3, 16,54,2,1, 16,56,1,4, 16,57,2,1, 16,59,1,4, 16,60,2,1, 16,62,1,5, 16,66,1,3, 16,67,1,1, 16,68,1,5, 16,69,1,3, 16,70,1,1, 16,71,1,5, 17,26,1,29, 18,3,1,30, 18,24,1,36, 18,44,1,31, 20,3,1,3, 20,4,25,1, 20,29,1,5, 20,31,1,3, 20,32,1,1, 20,33,1,4, 20,34,2,1, 20,36,1,5, 20,45,1,32, 20,66,1,3, 20,67,1,1, 20,68,1,5, 21,3,1,33, 22,20,1,3, 22,31,1,21, 22,38,1,5, 22,44,1,34, 24,4,1,3, 24,5,33,1, 24,38,1,5, 24,46,1,35, 24,55,1,3, 24,56,2,1, 24,58,1,4, 24,59,2,1, 24,61,1,5, 24,63,1,3, 24,64,2,1, 24,66,1,4, 24,67,2,1, 24,69,1,5, 0,0,0,-1

1010 REM CREATE "PAGEFOUR", ETC -- THE FILE WHICH
1020 REM ALLOWS USING A "PAGE" FOR OPERATOR INPUT OR
1030 REM OUTPUT TO THE OPERATOR OF DATA.
1040 REM --------------------------------------------------
1050 REM
1060 REM SINCE "CREAPG" IS USED AS THE NAME OF SEVERAL
1070 REM PROGRAMS--BE CAREFUL OF THE PAGE NUMBER!!
1080 REM
1090 REM
1100 OPEN "O",1,"B:PAGEFOUR"
1110 P1=1
1120 WRITE #1,CHAR*(P1)
1130 IF CHAR*(P1) = "*" THEN 1160
1140 P1 = P1 + 1
1150 GOTO 1120
1160 P1 = 1
1170 WRITE #1,LOCAT%(P1)
1180 IF LOCAT%(P1) = -1 THEN 1210
1190 P1 = P1 + 1
1200 GOTO 1170
1210 P1 = 1
1220 WRITE #1,DRW%(P1)
1230 IF DRW%(P1) = -1 THEN 1260
1240 P1 = P1 + 1
1250 GOTO 1220
1260 CLOSE 1
1270 END
10 REM STARTUP.BAS 8-14-84 JOHN C. DEBO
20 REM REVISED 11-4-84 ADD DATE & INVISIBLE PASSWORD
30 REM
40 REM
50 REM
60 REM ---------------------------------------------
70 REM STARTUP IS CALLED BY THE AUTOEXECUTE BATCH
80 REM FILE, TAKES CARE OF THE PASSWORD CHECK, AND
90 REM THEN PASSES CONTROL TO THE MAIN MENU IN
100 REM EXECUTIV.BAS.
110 REM ---------------------------------------------
120 REM
130 REM TO CREATE THE AUTOEXECUTE BATCH FILE
140 REM USE THIS PROCEDURE (IN MS-DOS)
150 REM
160 REM COPY CON: AUTOEXEC.BAT
170 REM ECHO OFF
180 REM BASICA STARTUP.BAS/F13/S;512
190 REM THEN ENTER (F6) AND (RETURN)
200 REM
210 SCREEN 0,0,0
220 KEY OFF
230 CLS
240 # = ""
250 REM
260 PRINT;PRINT;PRINT SYSTEM DISK --- UCF PLACEMENT CENTER"
270 PRINT" STAFF USE ONLY"
280 PRINT" TYPE PASSWORD PLEASE "
290 PRINT" THANK YOU. NOW PLEASE ENTER TODAY'S DATE"
300 PRINT" EXACTLY LIKE THIS--- 12-07-41"
310 PRINT" TODAY'S DATE==->";
320 INPUT"",#,M
330 IF M < 20 OR M > 99 THEN 430
340 DATE# = #
350 REM
360 REM THE DATE WAS CHECKED TO BE SURE IT IS BETWEEN
370 REM 1980 AND 1999 AS A PRECAUTION--BASIC WILL NOT
380 REM ACCEPT DATES EARLIER THAN 1980.
390 REM
400 REM
410 PRINT;PRINT;PRINT
420 PRINT" NOTE THAT THE PASSWORD HERE CAN BE ANY COMBINATION
430 PRINT" OF CHARACTERS UP TO 255 BUT YOU MUST SET THE
440 PRINT" LENGTH OF IT IN THE WHILE STATEMENT.
450 REM
460 PRINT;PRINT;PRINT
470 PRINT" IF # <> "DEBO" AND # <> "debo" AND # <> "1066" THEN 230
480 REM
490 REM
500 REM
510 REM
520 REM
530 REM
540 REM
550 REM RUN "A:EXECUTIV.BAS"
EXECUTIV.BAS Listing

LIST
10 REM EXECUTIV.BAS NEW ON 8-12-84 JOHN C. DEBO
20 REM REVISED 11-11-84 TO INCLUDE UPDAY
30 REM REVISED 11-22-84 TO INCLUDE MSGCREA
40 REM REVISED 2-10-85 ADD CHOICE & YESNO
50 REM
60 REM
70 REM
80 REM
90 REM

100 REM EXECUTIV CONTAINS THE MASTER MENU AND IS CALLED BY THE
110 REM START UP PROGRAM. ALL THE FUNCTIONS AVAILABLE ARE
120 REM ACCESSIBLE (CREATING AND UPDATING STUDENT AND EMPLOYER
130 REM FILES, CREATING AND REVISING INTERVIEW SCHEDULES, WRITING
140 REM REMINDER LETTERS, ETC).
150 REM
160 SCREEN 0,1,0,0
170 COLOR 6,0
180 KEY OFF
190 CLS

200 PRINT:PRINT:PRINT
210 PRINT" WELCOME TO THE UNIVERSITY OF CENTRAL FLORIDA PLACEMENT"
220 PRINT" CENTER INTERVIEW SCHEDULING AND STUDENT RECORD KEEPING"
230 PRINT" SYSTEM. THESE FUNCTIONS ARE AVAILABLE..."
240 PRINT:PRINT
250 PRINT" 1) START THE DAY BY UPDATING INTERVIEW FILES"
260 PRINT" 2) EDIT OR CREATE NEW STUDENT FILES"
270 PRINT" 3) EDIT OR CREATE NEW EMPLOYER SCHEDULES"
280 PRINT" 4) SEND A LETTER TO STUDENTS WHO MISSED AN INTERVIEW"
290 PRINT" 5) PRINT OUT A SCHEDULE OF EMPLOYERS INTERVIEWING"
300 PRINT" 6) EDIT OR CREATE MESSAGES FOR THE STUDENTS"
310 PRINT" 7) ENTER 'BASIC' FOR EDITING PROGRAMS & "
320 PRINT" 8) RUN THE STUDENT SERVICE ROUTINE & "
330 PRINT" 9) RUN THE FILE TESTING PROGRAM & "
340 PRINT:PRINT" * INDICATES YOU MUST RUN 'EXECUTIV' (DRIVE A) TO RETURN TO S
350 PRINT:PRINT:PRINT" SYSTEM "
360 CHOICE = 9
370 GOSUB 1170
380 PRINT:PRINT" NUMBER ";CHOICE;", ARE YOU SURE? (Y/N)"
390 GOSUB 1000
400 ON YESNO GOTO 410,240
410 ON CHOICE GOTO 700, 450, 500, 550, 600, 860, 750, 910, 960
420 REM
430 REM
440 REM
450 RUN "B:STDNUP.BAS"
460 END
470 REM
480 REM
490 REM
500 RUN "B:EMPLUP.BAS"
510 END
520 REM
RUN "B: DINGLTR.BAS"
END
RUN "B: DAYSCHED.BAS"
END
RUN "B: INTSCHED.BAS"
END
RUN "B: UPDAY.BAS"
END
RUN "B: STDNSRV.BAS"
END
RUN "B: FILETEST.BAS"
END

SUBROUTINE YESNO
-----------
THIS SUBROUTINE HANDLES YES/NO QUESTIONS, RETURNING YESNO = 1 FOR YES AND YESNO = 2 FOR NO
SOUND 523, 3
WHILE LEN(N$) < 1
N$ = INKEY$
1120 IF N$ = "N" OR N$ = "n" THEN YESNO = 2:RETURN
1130 IF N$ = "Y" OR N$ = "y" THEN YESNO = 1:RETURN
1140 SOUND 100, 4
1150 GOTO 1080
1160 REM -------------------------------
1170 REM SUBROUTINE CHOICE
1180 REM ----------------------------------
1190 REM THIS SUBROUTINE HANDLES CHOICES OF
1200 REM MORE THAN ONE OPTION. CALL IT WITH
1210 REM CHOICE = NUMBER OF LARGEST OPTION; IT
1220 REM RETURNS WITH CHOICE = NUMBER OF OPTION
1230 REM CHOSEN.
1240 REM
1250 SOUND 523, 3
1260 N$ = ""
1270 WHILE LEN(N$) < 1
1280 N$ = INKEY$
1290 WEND
1300 N = FIX(VAL(N$))
1310 IF N < 1 OR N > CHOICE THEN SOUND 100, 4: GOTO 1260
1320 CHOICE = N
1330 RETURN
Ok
BIBLIOGRAPHY


