Computer Science and Systems Analysis

Computer Science and Systems Analysis

Technical Reports

Miami University *Year* 1993

Implementation of the Interview Scheduling System for Miami University's Career Planning and Placement Office

Steven Peter
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1993-002

Implementation of the Interview Scheduling System for Miami University's Career Planning and Placement Office

Steven J. Peter

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

Implementation of the Interview Scheduling System for Miami University's Career Planning & Placement Office

by

Steven J. Peter
Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #93-002 March, 1993

Implementation of the

Interview Scheduling System

for Miami University 's

Career Planning & Placement Office

Steven J. Peter

Dept. of System Analysis
Miami University

Abstract

This project is an extension of systems analysis and design covered in two classes – SAN 472/572 (Analysis of Information Systems) and SAN 475/575 (Structured Design and Implementation). My implementation involved work in the areas of integrating a subset of the system, designing a multiuser accessing means, investigating database security, implementing dynamic calls, setting–up a database load facility, and researching laser printers and forms software. The integration involved getting the system to an operational state as opposed to a production state. Multiuser access to the system will be permitted to about 2600 Seniors and accounting Juniors. The students will be permitted to access the system simultaneously, and SQL will provide the concurrency control. The database security will be upheld by offering authority privilege to the program instead of the users. Dynamic calls will be used to maintain a reasonable virtual machine size. A database load facility will be used by the DBA at the beginning of every year to load demographic data pertaining to students. The standard resumes will be printed by the CPPO and that will require a new laser and a forms software. Research has been completed for various lasers and forms software, and a recommendation is given. The final portion of the paper presents some oft he benefits that I have obtained while working on this project.

Implementation of the

Interview Scheduling System

for Miami University's

Career Planning & Placement Office

Steven J. Peter

December 18, 1992

CONTENTS

Introduction

The interviewing process at the Career Planning and Placement Office is something with which I am familiar. After my undergraduate graduation my first job was obtained through the office, and while working toward my masters degree, I have used the facility to apply for internships. I have also come in contact with the CPPO through two classes which included a study on creating a computerized operation for the department. The courses were SAN 472/572 (Analysis of Information Systems) and SAN 475/575 (Structured Design and Implementation).

For my graduate project, I have made it my interest to expand on the work completed by these two classes. The project background and my accomplishments are presented in this paper.

Background

Richard Hearin, the Director of the Career Planning and Placement Center at Miami University, submitted a request for system services. He wanted to redesign the CPPO's current manual system. (For information of the current CPPO System, one should refer to the Placement Registration and On Campus Interviewing Procedures which can be obtained throught the CPPO.) But, in his request, he specified the current problems with the current scheduling sytem as follows:

1. Students place bids on companies that they would like to interview. To place a bid, the student must come to the CPPO which is a fifteen minute walk from central campus. These bids are placed within a twenty four hour period, exactly two weeks prior to the company's on-campus interview. Additionally, those students who win the right to an interview, must come to the CPPO to present their standardized resume (called the Placement Data Sheet) and sign up for the interview.

Both the bidding and sign-up process consume student time. For the vast majority of students, the trip to the CPPO is significantly out of their way. And in some cases, a sign-up can involve a lengthy wait in line because there are only two sign-up stations.

2. The current WANG processing system does not have any database capabilities; it is used in a word processing capacity to store some information on students and to record who is interviewing at given interview times. As a result, needed information cannot be readily obtained.

3. CPPO personnel spend much time determining who the winners are for interviews. The student bids, which are on 4x4 inch sheets of paper, have to be manually

sorted in descending order by bid amount.

4. CPPO personnel also spend time gathering the Career Planning and Placement Sheets for specific interviews. CPPO staff must manually put together a packet for each employer. And the whole process is spread over weeks. First the students have to bring in their data sheets, and the staff places them in folders. The day before the interview takes place, the packet is assembled. Up to thirty packets a day are constructed. They are needed at 8:30 A.M.

5. The personnel also gather the Career Planning and Placement Sheets for companies requesting information on students with certain criteria. There are approximately 2100 sheets to examine.

6. CPPO personnel are interrupted by students wanting to know their status with the CPPO or the bid points that they have left.

Rich Hearin indicated that he wanted a system which would enable the students to register with the CPPO, make bids, receive results and sign-up for interviews without having to physically come to the CPPO. In addition, he would like to be able to produce interview lists (a list of companies coming on a day) and copies of the Placement Data Sheet for each interviewer automatically. Finally, the new system should be able to to produce the data sheets of students who meet certain criteria.

In 1992, the students of SAN 472/572 approached this proposal and performed the system analysis on the project. The survey, study, and definition phases of the case determined that the project was a necessity and that the implementation of such a system was possible. Such a system should be able to perform the following events:

1. Registration and scheduling of employers (300/yr)
2. Registration of students with CPPO (2600/yr)
3. Student bid processing (15,000/yr)
4. Scheduling of interviews (300/yr).

The system should provide the following outputs:

1. Student registration and eligibility status
2. Recruiting Schedule
3. Interview & Requirements Schedule
4. Interview Bidding Results (listing of all bids submitted)
5. Master File Name Listing
6. Updated Master File Listing (sorted by Social Security Number with names)
7. Updated Master File Listing (without names)
8. Alpha Listing of Students (whose bids were accepted).

The beneficiaries of a new system will be both the students and the employees. A cost benefit analysis of a new system suggested that the benefits are intangible. The new system cannot result in any increase in income or decrease operation costs but may allow staff to dedicate work in other meaningful areas and will significantly save student time in the registration, bidding and sign-up process. Also, another intangible benefit is that "rush hours" relating to open sign-ups can be eliminated, and hence, work flow will be more evenly spread out for staff. Also, a new system will eliminate the human error that can occur in processing bids or selecting Placement Data Sheets.

The costs of a new system are more concrete. Any new developed system would have to be achieved on existing mainframe hardware. There will be an additional cost of a few personal computers and a laser printer. All and all, with the benefits and costs

determined, the SAN 472/572 class recommended that the project be implemented so that such improvements can be made in the interviewing process at Miami University.

Prior System Design

In the subsequent course, SAN 475/575, the design aspects of this case were considered. The database was designed using SQL. The entity relationship diagram for this design can be seen in Figure 1. Some screens were designed and implemented under XMENU. A system hierarchy of programs and modules was established. Programs were written in COBOL using embedded SQL and in most cases were designed using a similar module format offering familiarity for ease of maintenance.

There was no completed driver with this system; so these programs could not work in concert. Additionally, the system designed in this course was not a multiuser system which would allow many users to utilize the same applications and database at the same time.

Implementation

An Overview

My role in the project was to take the most important subset of the system to an operational state and also to research some special issues not yet addressed that would have profound effects on the system. The portion of the development cycle not yet complete was the integration of the separate modules into one larger module, the database construction and changes inside programs. The important issues not yet addressed for this system were multiuser access, database security, dynamic vs. static calls, laser printer and forms software purchase options.

My first work on the system was to integrate the modules written by the design class. Each module had been previously tested as a "stand alone" unit. But since the time that these modules were written, there were changes to screens, screen fields, the DFHCOMMAREA (the area where global variables are located) and to the database. So each module required detailed adjustments just to compile.

At the beginning of this semester, we had no active database in place. We had a database design from the previous class. I began to create the database that we had devised. It included five relations shown in Figure 1: STUDENT, EMPLOYER, SCHEDUL, INTERVIEW, and BID. These relations were required because of the embedded SQL that existed in all programs except for the driver.

My concentration was on six modules. Their relationship to the overall CPPO system can be seen in Figure 0.5. This figure shows the screen heirarchy of the system prototype and the previously mentioned six modules are related to these screens. The subset of the system that I concentrated on is indicated by circles on the figure. An overview of the six modules and their functions are explained in the following illustration and paragraphs.

Module Name	Description
CPPOCICS	is the driver.

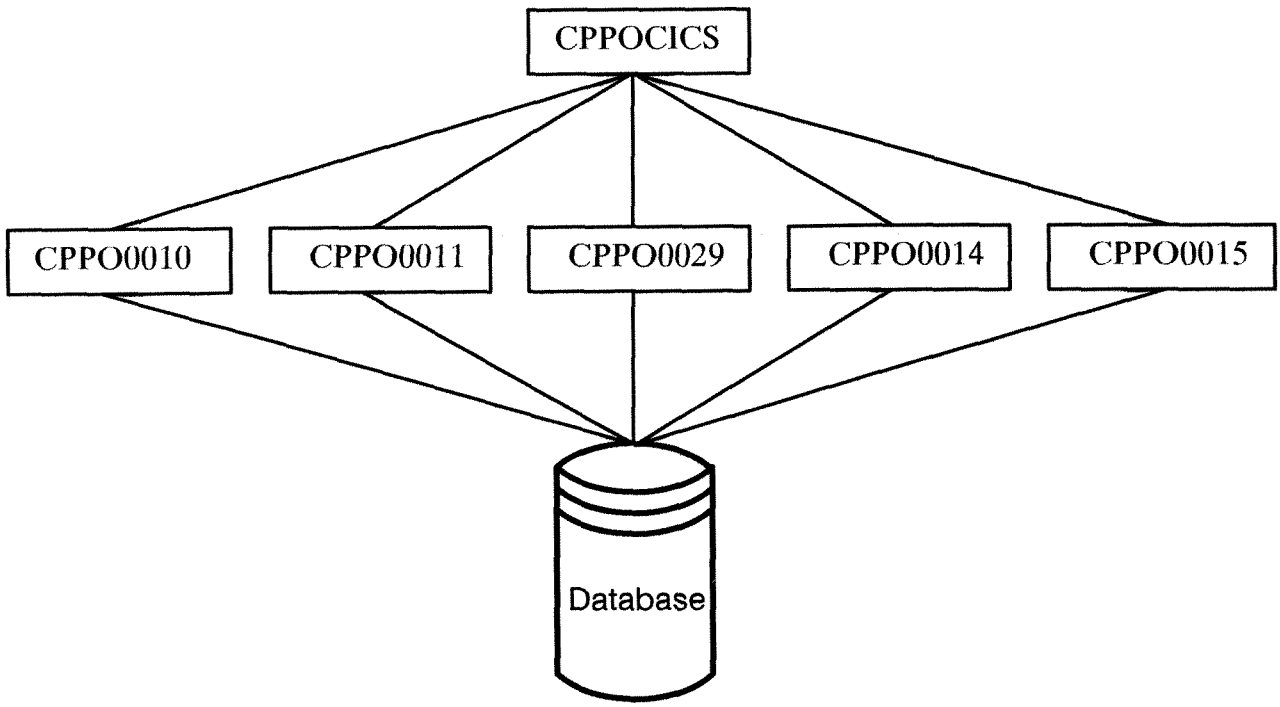
System Submodule Name	Description
CPPO0010	shows companies that the student is eligible to interview.
CPPO0011	produces more information about a specific interview; can place bid here.
CPPO0029	processes the bids for a company visit.
CPPO0014	presents the status of the student's bids.
CPPO0015	selects an interview slot.

The driver, called "CPPOCICS", contained no embedded CICS. Because of CICS's influence in industry, we wanted to use it. But we had to simulate true CICS because our operating system did not support it. The other modules were CPPO0010, CPPO0011, CPPO0029, CPPO0014, and CPPO0015. I will refer to these submodules as "system submodules" because there are other submodules that are associated with screen map presentation.

CPPO0010 is the module that presents the student with a listing of companies for which they are eligible by meeting their company requirements. The module's associated screen is SCRN1120. [See Figure 4.] Some changes were made to the fields of this screen. One improvement was to eliminate the individual fields across a screen line and make one screen line just one field. This reduces the amount of MOVE statements needed in CPPO0010; the whole record is moved to the screen field instead of many variables being moved to many screen fields. When the screen is displayed the user can select a given company to obtain more information about that company by entering the line number of the company of interest. By doing so CPPO0010 will move "CPPO0011" to the global variable, Next-Program-Name. Control goes back to the driver where CPPO0011 is called. Its associated screen is SCRN1121. [See Figure 5] Here, a student will see more information about an interview and will be able to place a bid.

Once the bids have been placed and the final bid date has passed, the bids can be processed. Module CPPO0029 processes the bids in the BID relation, revises the bid point fields (the amount that all bid winners are charged for a given interview) in the VISIT relation, changes the status of the visit, and updates the field in the STUDENT relation that holds the students remaining bid point balance. [See Figure 6 for the related screen and see Figure 17 for a before & after instance of certain relations] For a student to view the outcome of his bid, he would press the PF7 Key while at the screen, SCRNCICS. [See Figure 3] That action will take them to the CPPO0014 which will display SCRN1140. [See Figure 7] If a student becomes a bid winner, they can opt to proceed to the screen where they may select an interview slot. They enter the line number in the entry field, and the next program name is passed to the appropriate global variable. Control goes back to the driver where CPPO0015

will be called; CPPO0015 has the related screen - SCRN1141. [See Figure 8] Here the student can view the interview slots available for the company visit. They are able to select a slot and can even cancel a selection up to 9 A.M. the previous day of the interview.
A diagram of the relationship of my implementation is shown below.



The Integration

The driver was the first program that I updated. It did not have any embedded SQL so it did not rely on the database. Program name references in the calls to submodules had changed since the writting of this program. The DFHCOMMAREA had to be revised because each module was using different global variable names. The DFHCOMMAREA is the "01" level variable definition that the class used in the "Linkage Section" of our submodules. The name "DFHCOMMAREA" was used because it has significance in CICS. In CICS it is known as a "communication area" where the global variables are declared. In the system submodules, the class used the name DFHCOMMAREA to declare the variables that would enter the system submodule.

I also began work on the system submodules. In preparing the system submodules for compilation, the global variables between each system submodule and the screen submodule had to match. The screen module pushes the screen map to the monitor. Since there were name changes to screen fields and changes to the arrangement of the of related variables in working storage of the calling program, linkage adjustments had to be made. The code for the screen programs was produced by a code generator which would set the screen variables under DFHCOMMAREA. The record layout for the DFHCOMMAREA was fixed by the code generator. Therefore, it was easiest to change record layout of the system submodule.

After a successful compilation, the driver could not run with all options available until

all the called system submodules were in an executable state. Since the system submodules contained embedded SQL, the database had to be fully set-up to an operational state. Then there were changes to the system submodules. The changes to the system submodules were mostly in the areas of matching of the DFHCOMMAREA to the calling program, correcting the record layout of the global variables passed to the called screen program, an update of the embedded SQL, or as in one case, a total rewrite of the program logic. The rewritten program was CPPO0029. The transfer control portion of the driver had to also be written.

Once a module was completed, some verification of the program had to be performed. However, this testing was not extensive because it was not my objective to bring the modules to a production state but to bring them to an operational state. Under this verification, questions like the following were addressed. Did the queries select the correct data and display the information correctly from my limited number of test cases? With the screen displayed could the user page forward or backward? In most cases, there were quite a few corrections that had to be made. In the end, the package was operational which completed this phase of the project.

Database Load Facility

In testing the subset of the system that I had integrated, it became necessary to restore the database to its original state for retesting. Setting-up a reload facility for the CPPO database would also be important to the other student designers, and the technical support at the Computing Center. The support at the Computing Center will use the reload procedure to load demographic data that they obtain from the registrar to the STUDENT relation. This procedure will take place prior to the beginning of the academic year. The data that they will get from the registrar will exist in a sequential file format.

I have sent a copy of the STUDENT load facility to Bill Miley at the Computing Center for his use. At this time, all the attribute names are current, and the data in the data file has been laid out; so there is a guide for data placement in the sequential file. But everytime there is a schema change to the database, these load files will have to be updated.

Multiuser Access

Multiuser access was another issue addressed this semester. The question was, "how were we going to have multiple users access the CPPO System simultaneously?" In our virtual environment every "academic" user shares the level of the operating system known as VM. And each of these users will have an additional level of the operating system known as CMS in their virtual memory space. Similarly, the CPPO System would be resident in each CPPO user's virtual machine. And because of the many active copies of the CPPO System, we were also concerned about concurrency control.

There were four design options for our multiuser interactive system. The first option

involved a fixed number of of userids. The userids would be given names ranging from CPPO01 to CPPOnn where "nn" is the final number. A student who was interested in accessing the system would have to type one userid. If it is being used, he would try another until they found one that was open. The disadvantage of such a design is that it is clumsy and unprofessional. The advantage is that this design is easy to implement.

Another option was to have dedicated terminals set-up in various places. To access the system a user would sit down at the dedicated terminal. The student would just have to enter their personal identification number to the first screen of the system. Once again, this system would be somewhat easy to implement. Such a design would not require a change to the operating system. It would demand the creation of an active program that will monitor the dedicated serial ports and issue the system to those terminals. One disadvantage is that the system access is limited to those geographical areas where the dedicated terminals are located. And if the system goes down, a knowledgeable operator would have to be present to re-establish the system connection.

The third option involves a rotary style design. The underlying design is similar to that of option one, but the with the rotary feature. The rotary feature would allow users to logon to the system by issuing (at the userid prompt) one common userid that is associated with the system. There would be a maximum number of users that could access this system ranging from one to "nn". But the operating system would, behind the scene, find an available account so that the user would not have to attempt numerous logons as in option one. The major disadvantage with this design is that it necessitates a change to the operating system which can have unexpected occurrences.

The fourth option will be made possible by some policy changes made at Academic Computer Service (ACS). The new policy will be that all new students to the university will receive a userid that will remain theirs while they are registered. This userid will remain the same until the student graduates and no longer is registered at the university. The student would logon as he does now. First he issues his userid and password. Then he issues the class as "a" and the terminal type as "vt100." That will put him at the operating system command line at which he can issue the transaction name that will start the system.

This design becomes possible because ACS no longer will be as concerned with how many users are on the system at one time. The reason is that the maximum number of all-purpose users is roughly limited to the amount of terminals that the university has. In a meeting with John Kinney, Steve Moore, Dr. Schaber and myself, it was determined that if there are a great number of CPPO System users then this situation would be satisfactory and would not present a problem.

As for the concern about concurrency control of the database, SQL will provide the safety measures while in critical regions. SQL can multitask. Therefore, there is no concurrency control problem. The only other obstacle that we wondered about was whether there would be a possible database security breech by giving users access to the operating system command line. This topic is covered in the next section.

Database Security

Of major importance to those involved in the design and implementation of the CPPO System is the issue of database security. Students must be prevented from changing the database for their own benefit such as giving themselves more bid points. Additionally, students must be prevented from observing sensitive information about others.

At first, Steve Moore, Dr. Schaber and I thought that we may have some trouble in this area if we went with the fourth design option which would allow CPPO users access to the CMS command line. Our concern was that each registered CPPO student would have been granted authority privileges to access the system through one of the following commands: GRANT SELECT ON BID TO PUBLIC, or GRANT UPDATE ON BID TO PUBLIC.

Granting such privileges would allow a user to change the specified relation. But we discovered another way to grant privileges to the system, leaving user access from outside the system impossible. The solution is to grant privilege to the program; then only the program can examine or update the database. Such an example is GRANT RUN ON CPPO0010 TO PUBLIC. In this way, a user having the module CPPO0010 running in their VM space (and only under this condition) could indirectly access or update the database. Dr. Schaber and I tested this command from our DBA account. We then went into my account and ran the system which accesses the database. Embedded queries were allowed access. Then we exited the system and entered QMF where we tried to access the database. We were unsuccessful in accessing the database.

Once the GRANT command has been issued, it is a good idea for the DBA to check the meta-data associated with each relation. We are interested in who has been granted privileges for each of the relations in the database. This information can be found by issuing a query to the system table called SYSTABAUTH. The query is : SELECT * FROM SYSTEM.SYSTABAUTH WHERE GRANTOR = *'dba's id'* AND TTNAME = 'BID'. The result of the query may be:

GRANTOR	GRANTEE	GRANTEETYPE	SCREATOR	STNAME	TCREATOR	TTNAME	TIMESTAMP
SPETER	CPPO0010	P	SPETER		SPETER	BID	1992-11-20

The userid of the DBA would show under the GRANTOR field. The program was the recipient of the grant, and the GRANTEETYPE shows "P" which stands for program. A blank would indicate a user. TTNAME indicates the relation to which the program has privilege.

From this, the question arises, what is to prevent a student from issuing this "select" statement to find out what the name of a program is? And once they have the program name, what is there to prevent them from gaining access to the relation by writing a program using the same name with embedded code in their virtual machine. The answer is that a user will not be able to complete the preprocessing of the program unless the individual has been granted authority privilege to the relation.

Dynamic Vs. Static Calls

When I initially integrated the system modules I used a static link. As I added more system modules to the driver, the load time increased. [See Figure 9.5] With just six system modules and the associated screen modules, the load time was becoming significantly long. This consideration raised a concern as to what the results would be when the whole system was integrated. The overall estimate for system submodules is sixty. There was concern over what the effects would be on virtual machine size and load time. Of course the size of VM could be expanded, but with many CPPO users, a vast amount of memory would be required. I had to look into dynamic links.

With the new linkage, the driver would be dynamically linked to the submodules. But the submodules would retain their static link to their associated screen programs. There was no reason to call the screen programs dynamically since each system submodule usually possessed one screen module. The static linkage between a system module and one screen submodule uses a small portion of the virtual machine. This smaller load will allow the technical support staff to set a smaller virtual machine. The smaller the virtual machine, the more users can access computer services. Hence it is beneficial to have the system modules called dynamically, and the screen modules linked to the system modules statically. [See Figure 9]

So I investigated what was needed for dynamic calls. Recall, that all calls to system submodules were located in the driver, CPPOCICS. To convert the system to dynamic, two types of changes were needed. The first change was on the call statement itself. The second adjustment was a change in the compiling options of the compiler exec. After the code changes, each system submodule would have to be recompiled. At that time, the system would be ready to run dynamically.

A static call statement followed the format where the name of the submodule was bound by quotes in the CALL statement.

CALL *'program-name'* USING *a-list-of-variables*.

The dynamic call requires that the submodule name be moved to a variable name prior to the call statement. That variable would be present in the call statement. A CANCEL statement is used to release the memory used by the dynamically called module.

MOVE *"program-name'* TO *variable-name*.
CALL *variable-name* USING *a-list-of-variables*.
CANCEL.

Next, the compiling options needed a change. The options are located in the compiling exec shown in Figure 10. The fifty first line of the file is the crucial compiling line. This exec is set for a dynamic compile as shown by the "DYNAM" option. A static compile will have "NODYNAM" specified, or no specification at all, for which the default takes over as NODYNAM. With this alteration, the system submodules are ready for compilation.

With the linkages transformed, I thought it would be important to verify the dynamic memory usage. With a dynamic linkage, I would expect two results. First, the system execution should be slower since the called system submodules are not resident in memory. It takes time to load these modules into memory. Second, under a dynamic linkage, there should less virtual memory space used because the submodules are brought into memory when needed and discarded when complete.

To help provide information relating to the above assertions, there are some CMS commands that can be typed at the command line. To find out the size of a virtual memory,

the command STORMAP will provide this information under the VMSIZE field. Figure 11 shows the output result of this command. The size of virtual memory is four megabytes.

Another command is STDEBUG. This command can help monitor the amount of memory being used. To obtain the information about memory usage and time, the following commands should be issued as such:

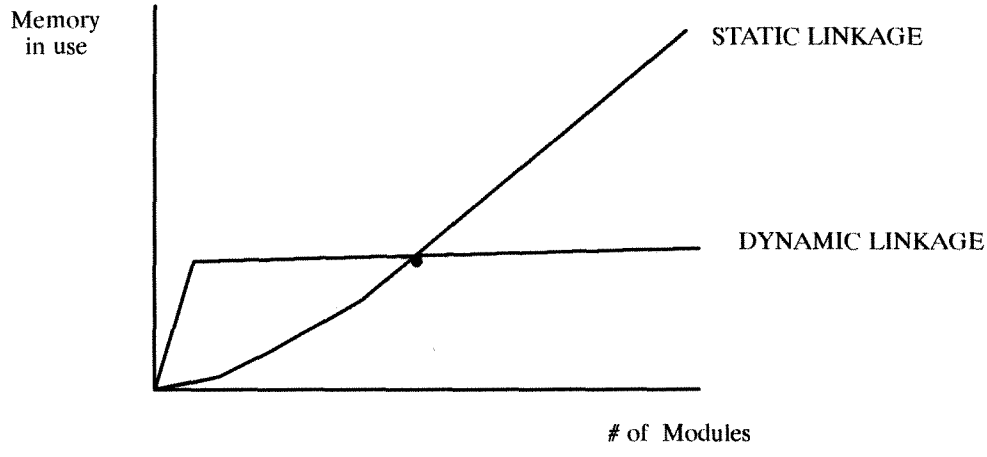
```
STDEBUG (OB REL FROM 0-40000 PUNCH D
LOAD program-driver-name
START
STDEBUG (END
SPOOL PUN CLOSE
Q FI
Q PUN ALL
TRAN PUN #### * R
RECEIVE #### filename
```

The first line initiates the action. The parameters specify that a record is to be created every time memory is "obtained" or "released." The second and third lines are typical commands used to initiate a program. For *program-driver-name* in the second line, I use the name of my driver - CPPOCICS. The program will run interactively, and the user can respond to the system. When completed and out of the system, the fourth line gets entered which will stop the memory monitoring process. The fifth line closes the punch. The sixth line finds the punch. The seventh issues a number. That number is used in the next line so that the output can be read into the user's reader. Finally the receive statement will allow the output to be placed in a named file. For more details on these commands, type HELP *command-name* at the operating system command line. "HELP" can assist with syntax and other options.

The static and dynamic output can be seen in Figures 12 and 13. Figure 12 is the output for the static run. Figure 13 is the output for the dynamic run. Column one of the output gives the time of the run. The same procedure was used under both static and dynamic runs. There were no delays between keystrokes which could add time to a test run. The procedure was to load the CPPOCICS module and respond to the driver screen by hitting the PF4 key which calls the CPPO0010 module. As soon as the companies were displayed, a "1" was typed in the entry area, and the PF4 key was pressed. Control goes back to the driver and the next program is called - CPPO0011. When the next screen was displayed the back-out key, PF3, was hit until I was out of the system. Total time is equal to ending time minus beginning time. As expected the dynamically linked modules took more time.

The next column of output specifies when memory is obtained or released. The amount of memory obtained or released is shown in bytes in the third column. At first look, the results were not what I had expected. I had observed that the dynamic linkage was using more memory space. Since then I have learned that there is an associated overhead with dynamic calls in CMS. If you look at Figure 13, you will see that early in the run their are some large amounts of memory obtained. The bytes in hex are 2000 (8192 in decimal) and 2010 (8298 in decimal). These large acquisitions are not found in the static run, but there are some similarities. The first four byte acquisitions in the static run can apparently be found in the dynamic run in lines four, nine, fourteen, fifteen and sixteen. But nothing can really be ascertained from this test about memory usage, because the static load is so small that the static results are not realistic. This static load consisted of six system modules and the screen modules associated with them. This would be a very small percentage of the real CPPO System

once it is complete. With a load of hundreds of modules, the memory usage should be much greater than the same system set-up dynamically. And the large dynamic system should have results similar to the small dynamic test that I ran. An illustration of this explanation is given below.



Hence there is an associated fixed cost of dynamic linkage and a breakeven point in the relationship between static and dynamic linkage. In my case, the number of modules fell quite short of this point, and the dynamic memory use exceeded the static memory use. But in the finished CPPO System, the dynamic memory use should be quite less than the static.

Aside from this test, there is another way to verify that the dynamic linkages did take affect. And that is to examine the load map once the dynamic driver has been loaded. The map should not list the system submodule system names (CPPO0010, ect.), whereas the static load map will have these program names. The two load maps can be examined in Figures 14 and 15. Since the dynamic load map did not have this submodules listed, the dynamic linkage was implemented correctly.

Desktop Laser Printer

Another area of the project that needed some research was the printing capabilities of the CPPO. Given that the CPPO was going to print the Student Placement Data Sheets, there had to be a way to print the standardized resumes. I investigated printer possibilities.

In addition to the necessity to print the standard resume forms, the CPPO also needs a laser that has the capability of printing clear text and graphics in a reasonable speed and forms capability. A reasonable speed is six pages per minute. The laser must have enough memory to hold the form image during the merge of the data and form. And lastly, it should have an interpreter program that can work with the most common page definition languages.

In consideration of these requirements, I found that the desktop laser, as opposed to the larger faster network printers, is what the CPPO needs. The desktop is perfect for the needs of a small work group such as the CPPO. Also for the user who has high volume of printing, the desktop can be set up as a dedicated printer.

There are several lasers to consider in this category. In November 1992, PC Magazine came out with its listing of mainstream lasers, those that print four to ten pages per minute. The six top lasers with their list prices that they recommend are:

Fujitsu Print Partner 10	$1,995,
Kyocera Ecosys a-SiFS-1500A	$2,395,
IBM LaserPrinter 10P	$3,795,
NEC Silentwriter Model 95	$1,749,
HP LaserJet4	$2,199,
HP LaserJet4M	$2,999.

The Fujitsu Print Partner 10 offers great speed at 10.1 ppm for text output. This laser doesn't have the complete network options that the HP's and the IBM has. Its base memory configuration is 1MB of RAM. It does have the ability to automatically switch fonts without having to switch a cartridge. The output is only 300 dpi (dots per inch) which is not that competitive in the market considering several of the above lasers can print at 600 dpi. And on a resume form, quality is important.

The Kycocera Ecosys a-SiFS-1500A offers low costs per page and supports environmental concerns by offering a drum that never needs replacement. Additionally, to assist with the environment, there no longer is the replaceable toner cartridge. Just the toner powder is replaced. The printing resolution is 300 dpi, which is not as sharp as needed for a resume form. The standard printer without upgrade comes with 1 MB of RAM which is less than some other laser printers.

The IBM LaserPrinter 10 was last year's "Editor's Choice." This year's IBM possesses many of the same features. The IBM is provided with 2 MB of RAM. It uses the Adobe PostScript as its standard font. It is possible to switch font automatically. It has a parallel port which is faster then a serial port because it sends one byte at a time as opposed to bit at a time. Most PCs have both a parallel port and a serial port. The inclusion of a parallel port enables the IBM to offer an automatic shut-off when the bin is full. In terms of text output speed, it is quite fast at 8.8 ppm. Of course, with graphics, the output rate would be slower, but the ability to print complex graphics is not pertinent, since the main need of the laser at the CPPO is for the printing of Placement Data Sheets. The output is top quality; it is produced at 600 dpi. But the HP lasers are slightly better when it comes to print resolution.

The NEC Silentwriter Model 95 has the lowest list price among these six that have been recommended. The standard printer comes with 2 MB of RAM. It prints at 6 ppm. The resolution is 300 dpi. It does support the page definition languages - Adobe PostScript Level 2 and PCL 5. It also has a fax option. But it is important to have resolution at 600 dpi.

The HP lasers use an improved version of PCL 5 which allows for 600 dpi printing (which is more important for graphics printing). They also have a new Canon engine (model P-270), the introduction of a RISC processor (to increase speed by reducing the number of instructions), and a quick parallel port which speeds up communication in both directions. Included with the HP lasers is a second paper tray. There is thirty five Intellifonts and ten TrueType fonts. The LaserJet 4M includes true Adobe PostScript Level2. Also included is HP's Resolution Enhancement Technology (RET) on both lasers. The 4M model has 4 MB of RAM. One changed added this year is the ability to change emulations automatically. Older HP lasers required various cartridges to be inserted to get PostScript fonts or some other desired font. LaserJet 4 is about $200 less than the LaserJet III which printed at 300 dpi. The performance of the HP laser on a small or medium sized network is good. HP does present the network option of JEtDirect network

interface cards for Ethernet and Token-Ring networks.

Reliability of a printer is of major concern. PC Magazine surveyed 18,130 of its readers and found the following results.

Manufacturer	% who had no repairs
Brother International	57
Canon,USA	58
Citizen America Corp.	62
Eastman Kodak Co.	60
Hewlett-Packard Co.	50
Epson-America	58
Panasonic	66
Star Micronics	67
Tandy Corp.	55

Although, these results pertain to all types of printers, they can be used in a general fashion to ascertain the reliability of laser printers. Note that the Hewlett-Packard registers on the list. None of the other recommended laser manufacturers are shown on the table. This is a positive attribute for the determination of what laser should be purchased. Also in Hewlett-Packard's favor is its new high resolution.

The reliability and quality of the printed output, causes me to recommend a HP laser in the four series. The HP's produce output at eight ppm. Hence, it will take the CPPO 65 minutes to print the daily Placement Data Sheets (based on 20 interviewers a day, each with 13 interviews and two data sheets per interview). Finally, retail prices are much less than list; an HP LaserJet4m can be purchased for less than $1,800.

I feel that the CPPO will need to purchase just one laser. They have a relatively new laser that is one year old, and it can be used when the newly purchased model breaks down. This model is Hewlett-Packard's LaserJet III P. Its quality will not be as good because it prints at 300 dpi. But for a back-up situations, this quality can be acceptable for a short term.

Forms Software

We wanted to keep the original design of the Placement Data Sheet, so an Electronic Forms Package will be needed to store the form and various "form letters." As for the Placement Data Sheet, the data relating to the student can be combined with the form to produce the finished Data Sheet that can be presented to the employer. The CPPO will need to print two data sheets for every interview on campus. Upon the request of an employer, the CPPO will also need to print data sheets of students that meet certain requirements.

So what is needed is a forms software package that will allow the CPPO to automate the job of taking the student data from the database (or downloaded database) and combining it with the form to produce the data sheets. More specifically, the form design is stored in the laser printer memory. A crucial part of this relationship is the laser printer itself because of the need to store the form in its memory. With the HP LaserJet, the PCL codes (that outline the details of the form such as lines, boxes, ect.) are resident in memory and sent to the output as a macro overlay. This overlay gets printed over any data sent to the printer. Since this overlay remains present in the laser memory and does not have to be downloaded to the printer each time, it actually takes less time to print repetitious, completed forms.

In the November 12, 1991 issue of PC Magazine, five forms software packages are presented and compared.
They are:

Form Publisher with Fill & File	$195,
FormGen Plus	$279,
Form Base	$495,
JetForm-Design	$495,
PerFORM Pro	$495.

Form Publisher with Fill & File is the most inexpensive of the reviewed group. It offers a complete assemblage of design tools, but the product is unsophisticated. One thing nice about the product is that simple forms are easy to create. With such form creation there is not a great amount of complexity placed on the user. But copying or creating an existing form is a difficult function, that takes a long time. The copying of such a form may require some concessions. The software does allow for the form design to be seen on the monitor. One disadvantage is that the product has this tape measure device which is only in inches. It is hard to set many lines per inch, and "clicking" on small objects is difficult. As for printing speed, the product upholds an adequate printing speed. The major disadvantage is that the form may not print as aligned on the monitor or some options selected in design may not show. And hence, it is these type of deficiencies relating to this product that warrant the non-consideration of this product for the CPPO.

FormGen Plus is also on the inexpensive side but it does the basic job. According to the editor, the attempt of creating a complex form for them cost several days, and in the end they could not produce the form. Also once a form is saved, screen-lockouts are put in place, and the form cannot be changed. Different thickness of lines cannot be used in a form. The product allows for quick printing, but it is the troublesome creation characteristics that prevent the recommendation of this product.

FormBase is good in the area of database management; it has its own proprietory

database. Data can still be imported from ASCII text though. It drawing abilities are limited. There are no tools for drawing; resizing cannot be done, and complex forms are difficult. Printing production is at the low end of these reviewed products Also, a company helpline is offered for only the first 60 days after purchase. After that, the user must purchase a service contract which costs $100 for individuals or $1,000 for corporations. The other option is to call the 900 line which costs fifteen dollars per call. The lack of support for the customer is a real disappointment for this product.

Of the five, PC Magazine recommends two – JetForm–Design and PerFORM Pro. The two offer many of the same features. Both have a screen design features and further, each will accept a form that has been entered via scanning device, a feature that may be useful as a future enhancement to the system. Both are able to take care of data application. Both provide quick printing. Both products support a feature called forms fill–in, a feature that allows for data entry through the screen form image. The forms fill–in could be another future enhancement to the system. As for data entry, both products offer entry modes of character graphical. Each has the ability to verify data entered and to run calculations. It is possible to merge data and forms in batches with these software products. This process can be issued from the DOS command line. But for PerFORM to achieve this function there would be an upgrade with additional expense.

JetForm is also recommended but for large corporations with a multitude of form needs. The overall package consists of the combination of several software products. One product is known as JetForm–Design which is used in form creation, and another is JetForm–Filler which allows for the "fill–in" and printing of forms. The product does support some field validation rules, but these are not covered sufficiently in the manual.

The design features associated with this product are good. It facilitates quick and easy form design. The user can duplicate existing forms or objects. Small objects are manageable with the zoom feature. Measurement can be in several units: inches, millimeters ect. A wide range of fonts are available. One disappointment is that objects cannot be resized once drawn. JetForm does support network features by allowing the forms to be stored centrally. The forms can be used by many users whether their work is mainframe or minicomputer based.

Once a form is compiled, the process of printing the forms involves use of the Filler program, which combines the data and the form. The process is fast because once the form is in place (in printer memory) only the data needs to be sent to the printer. The data can be in ASCII format. The JetForm requires a 386–based PC with 2MB of RAM and 2.5 MB of hard disk space. It can run on releases later than DOS 3.1 or Windows.

PerFORM also has modular structure. This modular separation allows for user separation from the design facility. Other components of this product include its network accommodation, excellent form design facility, and a management database. PerFORM allows for the application of an encrypted user signature an interesting feature. As mentioned earlier, PerFORM also provides the form fill–in element. Forms can be printed from an ASCII file. But the product does not "have a built in script language." Apart from JetForm, it has a resize feature and the ability to move objects to the front or back. It has the potential to show or suppress boarders, file locking, database record locking, digitizer support, some additional graphic formats (EPS, GEM, .IMG), more predefined lines widths, arrows, shades and colors, pop–up query boxes and database indexes. It requires at least a 386–based PC and 2MB RAM with 3–5 MB of hard drive.

In conclusion, for the same price as JetForm, there are many more features included with PerFORM. On calling a mail order retailer, I have found out that PerForm is in version 2.1. Its cost is substantially less than the list price indicated. It is $183 plus eight dollars for shipping. I have recommended PerFORM for the forms software, and have completed the requisition forms for purchase. I expect to receive the product by the end of the semester.

Documentation

Throughout the course of the semester, I have had meetings with certain members of the CPPO and The Computing Center. The minutes for those meetings are contained in the following pages.

Benefits Obtained Through This Project

Looking back at the project, there were many things that I learned and things that were reinforced during this semester. While starting the project, I foresaw many of these benefits. But now at the culmination, I have found that there are a few more additions.

The first benefit was the experience of being able to examine other programmers coding style and to use some of their techniques to expand my own. Of course, under this system there was some limitation to the expression of style that a programmer could exhibit. The limitation was that all system modules were built with a similar structure. For instance, the procedure format was as follows:

```
0000-MAINLINE
1000-PREPARE-SCREEN-OUTPUT
2000-DISPLAY-SCREEN
3000-EVALUATE-PFKEYS
4000-TRANSFER-CONTROL
90100-GET-DFHCOMMAREA
90200-PUT-DFHCOMMAREA.
(*Ninety thousand level procedures contained the embedded SQL.)
```

Much of the code within these procedures is now being produced by a code generator; so it will be very similar in structure. And the structured format should aid in maintainability by offering recognizability which will help the programmer locate an area of the program that gets the update. Aside from this style limitation, there were still many coding styles to observe. To name a few, they were the person's use of two dimensional arrays, a coders's means of implementing a page "forward" or "backward" and the individual's use of certain COBOL commands such as the STRING statement, the EVALUATE statement, and form of the dynamic call.

Another benefit obtained was the experience of the integrating modules into one load. There are the standard issues reinforced that are related to COBOL linking. The calling program must have the passing variables declared in WORKING STORAGE but not in the LINKAGE SECTION. (The driver has no LINKAGE SECTION.) The CALL statement in the calling program contains the using clause which specifies the global variables.

```
WORKING-STORAGE SECTION
    .
    .
 01 DFHCOMMAREA.
     05 FIRST-50-BYTES          PIC X(50).
     05 SECOND-50-BYTES         PIC X(50).
     05 THIRD-50-BYTES          PIC X(50).
    05 FOURTH-50-BYTES          PIC X(50).

PROCEDURE DIVISION.
   MOVE 'CPPO0010' TO CALL-PROGRAM-ID.
   CALL CALL-PROGRAM-ID USING DFHCOMMAREA
   CANCEL CALL-PROGRAM-ID
```

The called program contains a LINKAGE SECTION with the passed variables

```
LINKAGE SECTION.
01 DFHCOMMAREA.
    05 FIRST-50-BYTES          PIC X(50).
    05 SECOND-50-BYTES         PIC X(50).
    05 THIRD-50-BYTES          PIC X(50).
   05 FOURTH-50-BYTES          PIC X(50).
```

The procedure division indication specifies the passed variables and the program ends with a GOBACK statement.

```
PROCEDURE DIVISION USING DFHCOMAREA.
    .
    .
    .
    GOBACK.
```

Another valuable piece of knowledge gained was that relating to the virtual machine. Creating a multiuser, interactive system on a virtual machine required some research. The important points that are unique to the virtual machine are that each user has their own designated portion of memory. These portions of memory are built on top of VM. CMS is duplicated in each virtual machine. Additionally, the CPPO System would be resident in the virtual memory space of the users accessing it. The CPPO System would be redundantly resident in memory as many times as there are users running the system.

Similarly, the multiuser system design under VM was a learning experience. Since the VM operating system is not designed to support a multiuser interactive environment, there were several options that had to be investigated for user access. Exploring those options was informative.

An additional positive experience gained while working on the project came from the use of the debugger. The tool increased my familiarity with such a tool and helped to streamline my approach for locating errors. Given an error, the experience has helped me anticipate where to look for the cause. Or when the cause was unknown, I could use the debugger to step through the code and check variable values. My approach to debugging was to locate pertinent data, organize it in the relations, study the relationships, devise a hypothesis, run the program, check the output, then use the debugger to find the error.

The project also allowed me to develop my fact finding skills. There are many means to discover facts but my technique involved interviews with members of the CPPO. The purpose of the meetings was to clarify matters, verify understanding, generate enthusiasm, get the end-user involved and to solicit ideas and opinions. The personnel were able to further specify what the system should provide. Some of the facts uncovered during this semester are shown in the appendix of this paper under the meeting minutes section.

One other benefit acquired from the project was the further development of my decision making skills. Some of those decisions included, adding a STATUS field to certain relations, recommending a forms software and laser printer, changing the logic of CPPO0029, deciding what variables would be needed in DFHCOMMAREA, ect.

Another experience gained was seeing how a design change or a change in specification can alter the progress of the project. The client has a tendency to want to add features to the system as the implementation progresses, and the implementer discovers facts not found in analysis. Overall, I found that a change took more time to implement once the system was being implemented than if the change in specification occurs before implementation.

Implementing this system has also helped me further develop my communication skills. Working with people during the project, holding a biweekly meeting, meeting with my advisor, expressing changes to be made, doing fact finding, and working with people in technical support to resolve issues were activities that helped in this area. Throughout the semester, I had meetings with Steve Moore in technical support, to address the issues of database security and multiuser access. I also meet with Bill Miley from The Computing Center and some of the CPPO Staff every other week to cover new issues and discuss system progress.

In the project, I was also able to expand my knowledge of SQL database security. I was able to learn that it was possible to grant authority to a program so that a user could not access the database outside the system. Likewise, I was able to query the previously unknown SYSTABAUTH table to find authority details.

In progressing through the development of the system, I had a great many opportunities to refer to system manuals relating to SQL, COBOL, CMS operating system, and HP laser printer. The experience will prove valuable as it has caused me to seek out answers on my own which will reduce my dependence on others in the work force.

Furthermore, I expanded my understanding of forms software as I had to research what such a product does. I had to review certain products so that I could make a recommendation on which product should be purchased for the CPPO system. The same procedure occurred for a laser printer.

The project also allowed me to gain experience in testing. Although more testing will be necessary on the portion of the system I integrated, I was able to administer black and white box testing on individual modules and the integrated modules. I was able to plan and set test cases and anticipate their effects.

Another benefit obtained while implementing this system, was the use of a screen painter, XMENU. I was able to further develop my understanding and skill using the tool. The knowledge will be applicable at work with other screen painters.

The final benefit that I obtained while working on this system was the mindset of creating a system that is user friendly. A usable system is one that is user friendly were the interface with the users has similar conventions on screens through out the system. Such work involved making sure that the cursor goes right to the field where input is required, having the same PF keys perform similar functions in various screens, having input fields containing under-

scores ect. This is an important issue when creating a system. It is one that I will stress when working in industry.

Conclusion

To this date, the implementation has been completed to the point set out in the proposal. That point is the integration of an important subset of the CPPO system. Additionally, some important issues relating to the system were researched, and solutions were developed to those integral pieces.

In addition to my efforts, the students from the design class, SAN 475/575, have managed to further design other parts of the system and implement them. The student undertaking has taken further the student side, staff side and administrative functions of the system. On the student side, they have completed in modular structure the programs that complete student registration. On the staff side, they have worked on the programs that register an employer, schedule an interview, take room inventory, customize interview slots, present standard slots, view schedules, display information about a student and list an employer. Student work on the administrative side has added to design the programs that print daily reports, print management reports, perform the end of year functions, change password, create daily interview packets, record "no shows", create custom packets, suspend or reinstate students and determine room availability.

Despite these undertakings, there is still much work to do. A list of items that I have noted is as follows:

-integrate the remainder of the system,
-perform extensive testing on the finished product,
-design the system subset that creates a diskette version of the Placement Data Sheet,
-get with the CPPO to decide what will be the abbreviations for MAJORS,
-give this information to Bill Miley so he can use it to transform registrar's demographic data into the database,
-in the student entry area, incorporate a validation on a student major change,
-in CPPO0011, allow for minimum GPA if accounting major,
-set Backup & Recovery procedures,
-complete documentation on each module and give to Bill Miley who will oversee maintenance,
-create the Placement Data Sheets on the forms software,
-provide a flag indicator to show when a GPA field has been changed from that given from registrar,
-double check with Rich Hearin that there will be no overflow list,
-change system so that when student cancels an interview (and all slots are taken) that the interview goes to open status,
-show Open sign-ups on screen SCRN1140 from CPPO0014,
-allow a student to be on schedule when they do not meet the requirements,
-place time limit in canceling interview in CPPO0015 (9 A.M. the previous day),
-place deadline in taking an open sign-up. (1 P.M. the previous day),
-allow MBA's to bid on undergraduate major that they had,

-allow interdisciplinary majors to bid on their "area of focus",
-allow CPPO staff the ability to move other students into the unfilled interview slots for another position. For example, a company is interviewing for a full time systems job. All the slots are taken for this position. But for its summer internship position, only half are taken. The company may allow or permit the remainder of these slots to go to full time job interviews.

I would anticipate that the project can be further implemented by the student classes. It would also be helpful if another graduate student would take up the development as their graduate project. There is a possibility that a group of undergraduate students could do the same as their senior project.

The implementation of this project will proceed because the customer evaluation of the system has been positive. The director of the CPPO, Rich Hearin, has seen the prototype, and he has said that he likes it. Biweekly meetings with some of the CPPO staff have been held to obtain their input to the system. They have also seen the prototype and like it. So the implementation should continue.

Appendix

The minutes, figures and displays are shown in the following pages.

a. Minutes of Meetings

MINUTES FOR MEETING DATE: 9/17/92
Concerning the CPPO

Present: Rich Hearin
Bill Miley
Keith Weber
Joe Barry
Dr. Tom Schaber
Steve Peter

The two main questions that we had prior to the meeting were, where will the system run and who can support it.

The meeting began with a discussion on the merging of ACS and TCC (the Computing Center).
What exists now is the University Information Systems, which oversees Administrative Systems and supports all other university systems activity. Instructional Systems assists faculty and how they use the media.

Early in the meeting we were informed that UIS could support COBOL and that the CPPO System should reside on the academic side of the mainframe that which supports the CMS operating system. These two pieces of information are favorable because much of our system design was based on the mentioned programming language and operating system. As for the database, we indicated that we would be using SQL/DS, and that Steve Moore has expertise with the product.

In the meeting, I had called attention to my graduation in December. Rich Hearin asked how much of the project will be completed by that time? The answer is as much as possible. The project will be fully implemented by August 1993. It will be necessary for another graduate student or another class of Dr. Schaber to take up the project next semester. Rich hopes that implementation does not carry into August; he would like to use and test it. He may "test market" it on System majors.

It was asked, what will be the accessibility to the system. I said that system obtainable through any terminal on campus, but most notably for students, terminals at Kreger and Hughes would be most popular. Add on features to the system may include an additional menu option at the networked microlabs. Rich Hearin mentioned that system may want to handle modem connection.

I brought up the issue of how can a interactive multiuser system be implemented in a CMS environments. A possibility was explained such that there could be multiple users on one account and that there would be a maximum of "N" number of accounts.

Rich Hearin asked how the Placement Data Sheets will be added to the database and whether that could be done by scanner entry. I mentioned that the students will have the data entered via screen fields in the interactive system. We agreed that the CPPO would like the output capability which would involve a laser printer. This output portion of the system has yet to be written. In addition, Rich remarked that the CPPO can save several thousand dollars a year if it can produce a diskette version of these Placement Data Sheets that they sell to employers. Currently the CPPO has to contract this service to an

outside firm; what they receive is the Placement Data Sheets themselves. Linda Smith handles the contact with this firm.

The issue of Security was brought to attention. Rich stated that he wouldn't want students being able to generate additional points ect; he wants the "fairness kept in" the interviewing process. We assured him that this was the most important point, and that we will test the system for such defects.

A question brought forth was, will there be help screens. Our answer is yes, but who will put the text in the system is another issue. Maybe we could receive some help from those at the CPPO.

Rich asked what should he be planning for his equipment. It was mentioned that someone form Debbie Allision's group could work with him on this. Hardware people can assist with interface considerations for the CPPO personel. They can offer trainning, problem resolution, and equipment repair.

I asked (relating to our database) if there were any problems in obtaining student demographic and grade data from the Administrative data that is held. The answer is yes, it is just a matter of getting the permission from the student to release the infomation. It would be satisfactory to have an entry field that says, "by typing your name here, you are giving your support to release this information." Such a feature will reduce the time it will take for a student to register in the system therefor reducing the number of people on the system, and it will also releave the Registrar from having to satisfy student requests for GPA information. The data would be downloaded from the administrative data to tape and input to the CPPO system database via tape.

We will have to determine how this demograghic and grade data will be obtained or added for transfer students.

A dropped CPPO system requirement is that of literature distribution. The system will still indicate that company literature is available and would be helpful during the interview. But we will no longer have to incorporate in the database a field indicating whether a student came by and picked up information about the company.

However, the "Prenight" must have a different key to acknowledge.

Bill Miley requested a Project Schedule.

Meetings: Initially, we'll have one every two weeks - Thursdays at 9 A.M. The next meeting is October 1, 1992.

The meeting concluded, and a tour of the mainframe equipment was given.

Time of this meeting: 2 hours.

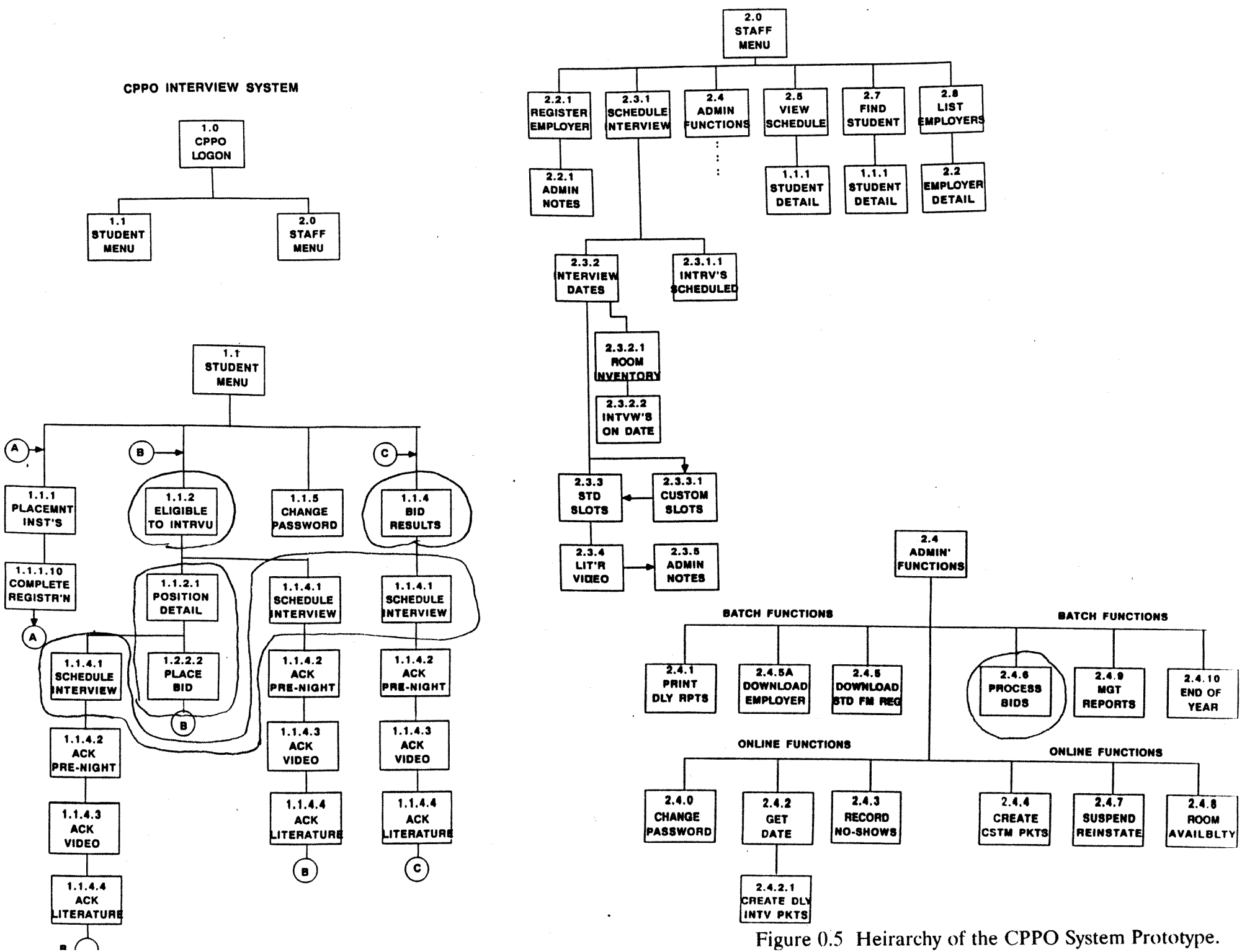


Figure 0.5 Heirarchy of the CPPO System Prototype.

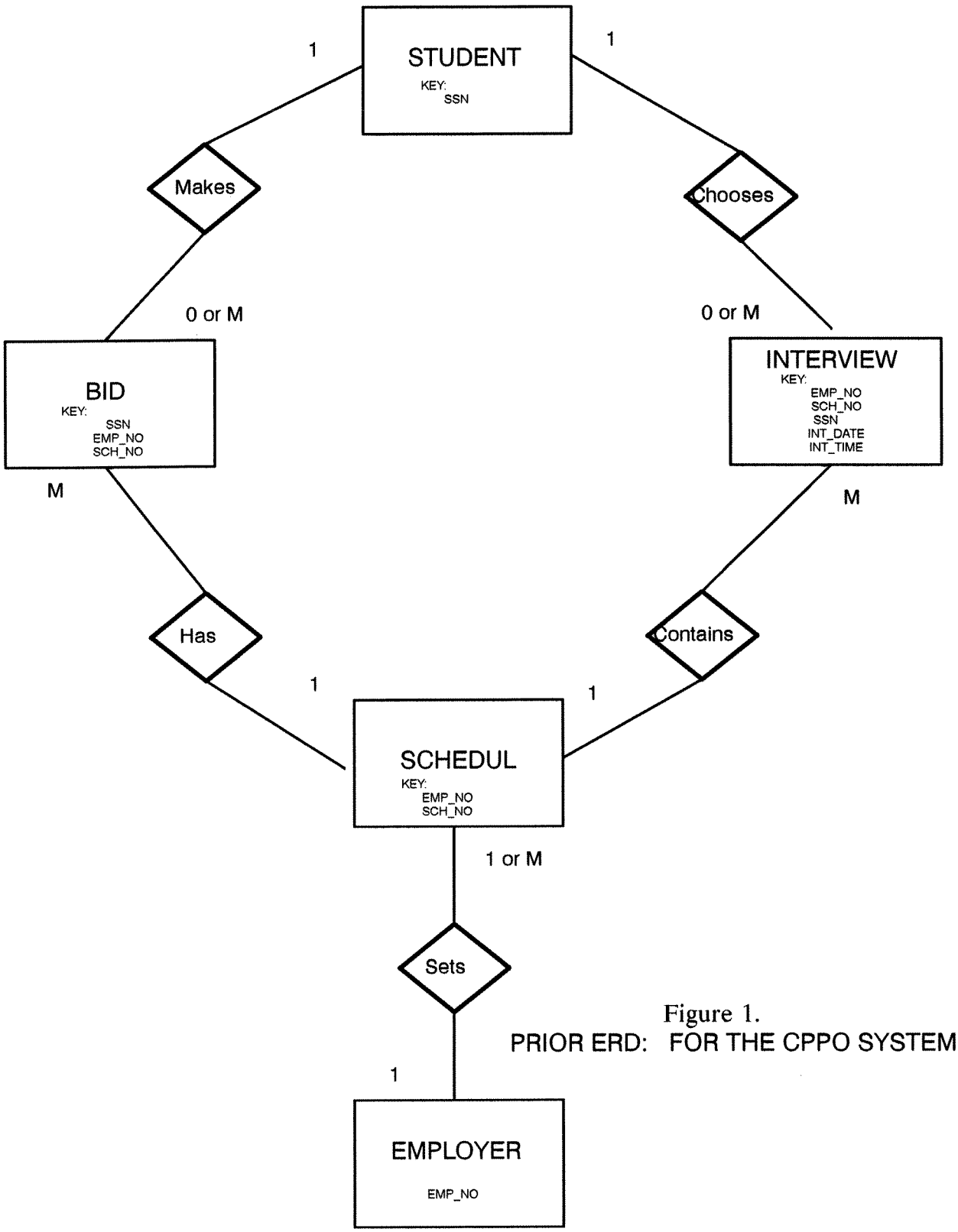


Figure 1.
PRIOR ERD: FOR THE CPPO SYSTEM

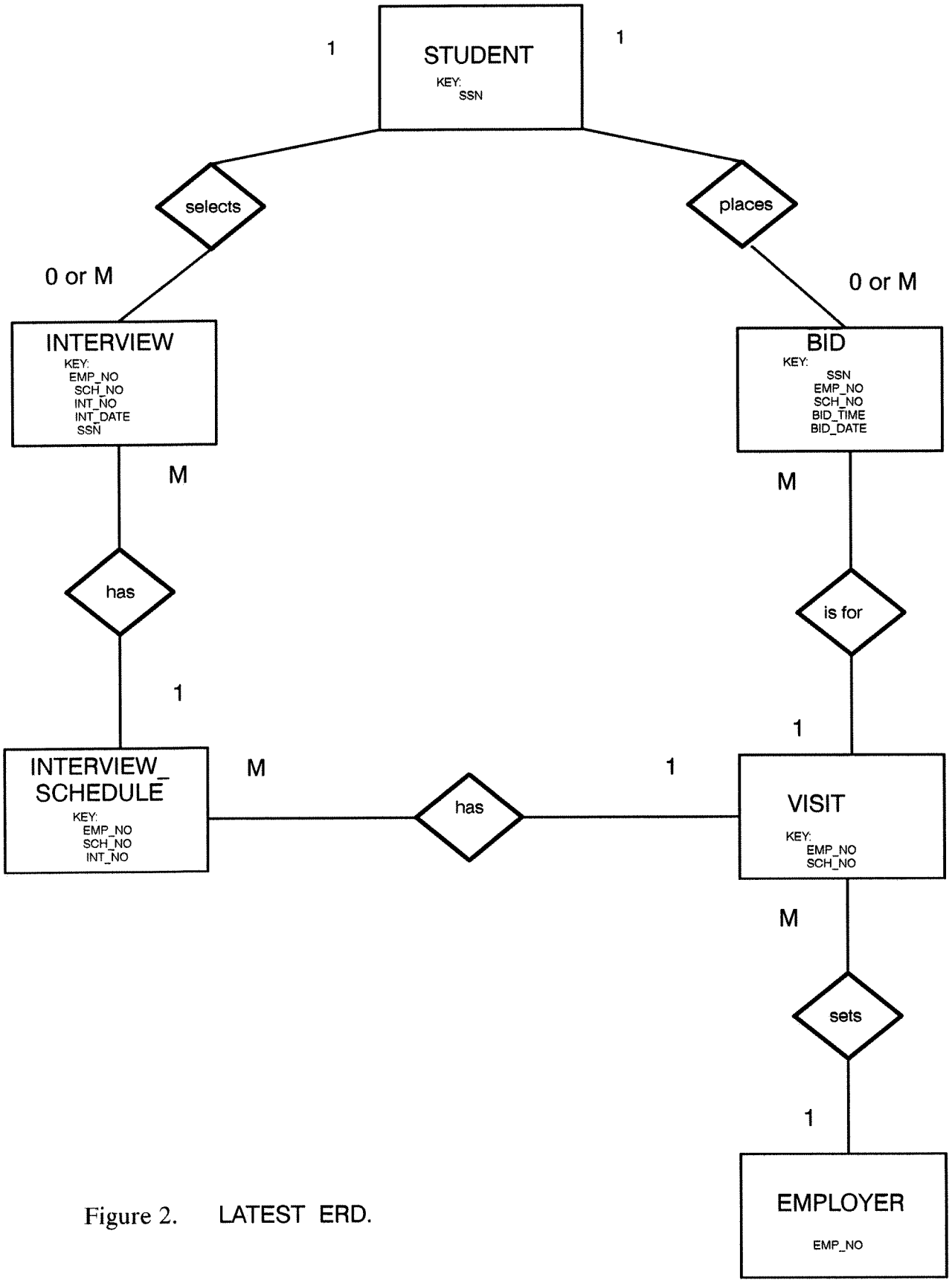


Figure 2. LATEST ERD.

```
                    TEST CPPO BACKBONE PROGRAMS
              Execute desired program by pressing indicated PF key.


./F1  - CPPO0038 - Display Help message for this screen (not implemented).
PF3   - CPPOEXIT - Exit this CPPO TEST system.
PF4   - CPPO0010 - Display Companies with whom this student can interview.
PF5   - CPPO0011 - Enter bid for selected employer.
PF6   - CPPO0015 - Schedule an interview for this student for this company.
PF7   - CPPO0014 - Display Bid Results for this student.
PF8   - CPPO0020 - Create the interview records for an employer's schedule.
PF9   - CPPO0029 - Bid Processing (Batch)
PF10  - CPPOWONE - ??????


DFHCOMMAREA (First 200 bytes of common data area; make changes if you wish).
 22222222201234567890123456789012345678901234567890
 12345678901234567890123456789012345678901234567890
 12345678901234567890123456789012345678901234567890
 12345678901234567890123456789012345678901234567890
 12345678901234567890123456789012345678901234567890
          1         2         3         4         5
```

Figure 3. Driver Screen.

CPPO STUDENT INTERVIEW SYSTEM

CPPO0010 SCRN1120 COMPANIES INTERVIEWING FOR 11/28/92 17:52

MAJOR MTH

The companies listed below are the companies you are eligible to interview based on your Degree, Major, and Graduation Date.

#	EMPLOYER	POSITION	LOCATION	STATUS	BID DATE MO/DY
1	NCR	STATISTICIAN	DAYTON, OH	BID	01/12
2	P & G	STATISTICIAN	Cincinnati, OH	BID	01/14
3	KROGER	STATISTICIAN	Cincinnati, OH	BID	01/14
4	GENERAL ELECTRIC	STATISTICIAN	Cincinnati, OH	BID	01/14
5	CINCINNATI GAS AND ELECTR	STATISTICIAN	Cincinnati, OH	BID	01/14
6					
7					
8					
9					

For more information about a particular employer or to place a bid, enter the employer number here: _ and press PF4.

PF1:HELP PF3:QUIT PF4:Detail & Bid on Selected Company PF7:PG BK PF8:PG FWD

Figure 4.

Eligible Position Screen.

```
                          CPPO STUDENT INTERVIEW SYSTEM
PPO0011  SCRN1121         PLACE BID FOR SELECTED EMPLOYER          11/28/92  17:58
         Employer NO:     999
              Sch No:     1
            Employer:     NCR
            Location:     DAYTON, OH
              Majors:     MTH                                Degree:    G
                                                          Grad Date:    M/93

        Interview Date:   01/26/93
              Bid Date:   01/12/93                 Number of Slots:    15

    Misc Notes:   ADDITIONAL REQUIREMENTS GO HERE!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 YOU CURRENTLY HAVE  5000  BID POINTS LEFT.
 You may place or change a bid anytime up to midnight of the above bid date.
 Bids may be changed any time by simply submitting another bid.
 Bids may be deleted by submitting another bid with 0000 bid points.
 Notice that bid entries must have leading zeros. Example: 0075.
 IF YOU WISH TO PLACE A BID, ENTER POINTS TO BID HERE: ____ AND PRESS PF4.
PF1:HELP    PF3:QUIT    PF4:PLACE NEW OR REVISED BID
```

Figure 5.

Position Detail and Place Bid Screen.

```
                        CPPO STUDENT INTERVIEW SYSTEM
PO0029   SCRN2460            RUN BID PROCESSING            01-13-93  17:59:51

The Bid Processing Program was last run on  01/12/93  at  18:52

It was initiated by  JOE CPPO

Press PF4 if you wish to process the latest group of bids.
Bids processed in previous runs will not be processed again.

Press PF3 if you DO NOT wish to run the Bid Processing Program.




PF1:HELP  PF3:QUIT   PF4:Run Bid Processing Program
```

Figure 6. Bid Processing Screen.

```
                        CPPO STUDENT INTERVIEW SYSTEM
 CPPO0014  SCRN1140                BID RESULTS                 11-28-92   18:01

 STUDENT ID:     222222222                      REMAINING BID POINTS:  5000
 STUDENT NAME:   EDWARD M BERK

                              WIN'G     YOU
 #   EMPLOYER                   BID     BID   _____________________STATUS_____________
 1   NCR                       0900    0900   SCHEDULE AN INTERVIEW BY 1993-01-15 .
 2
 3
 4
 5
 6
 7
 8
 9


 SCHEDULE INTERVIEW WITH EMPLOYER #: _ ; PRESS PF4.

 F1:HELP   PF3:QUIT   PF4:Schedule Interview
```

Figure 7. Bid Results Screen.

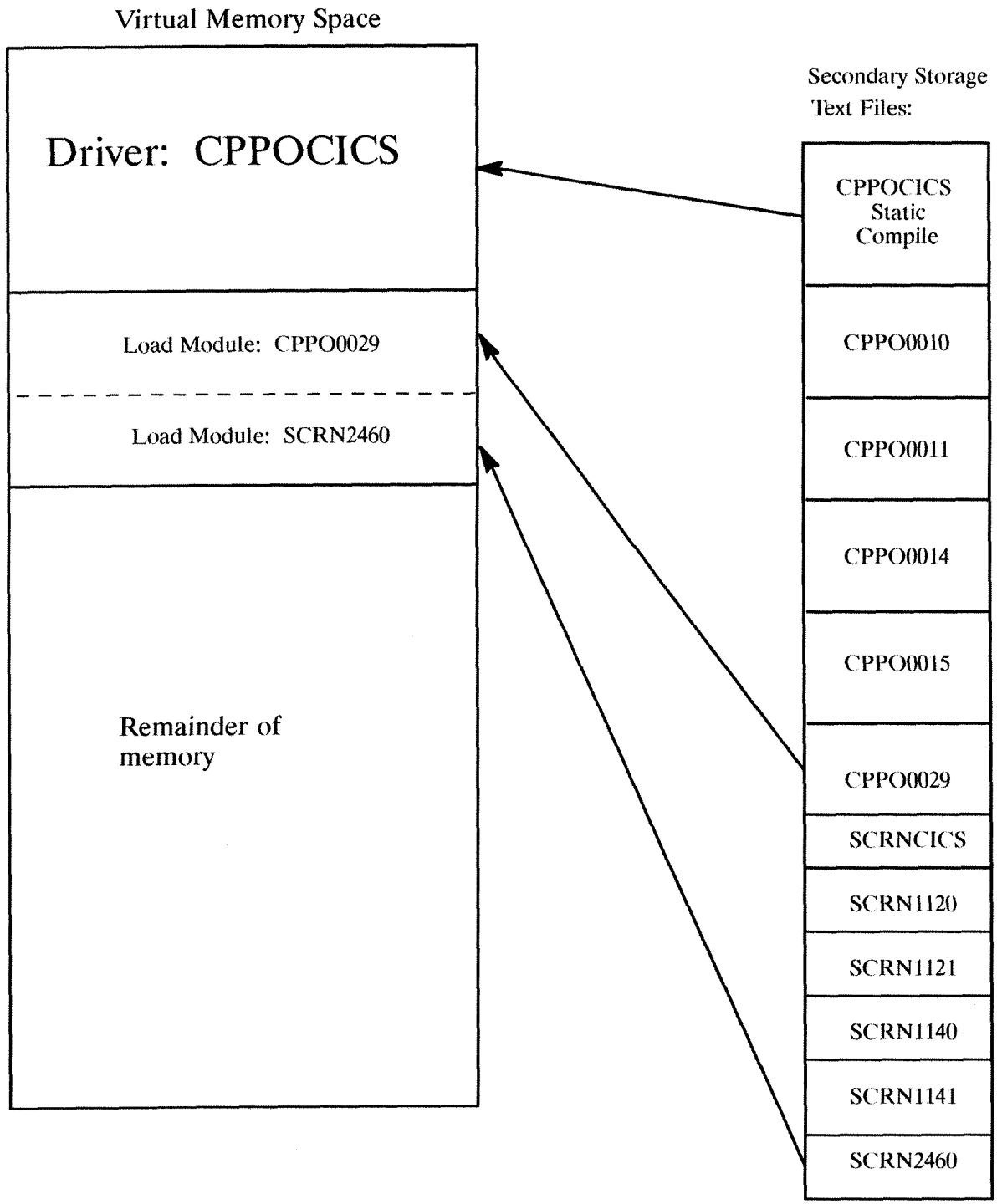
```
                    CPPO STUDENT INTERVIEW SYSTEM
                       SCHEDULE YOUR INTERVIEW          11/28/92   18:01:39

STUDENT ID: 222222222   SCHEDULE:    999 -  1     NCR

 MORE
 SLOT       DATE          TIME             STATUS
 01       01/26/93        08:00            TAKEN
 02       01/26/93        08:30            TAKEN
 03       01/26/93        09:00            TAKEN
 04       01/26/93        09:30            TAKEN
 05       01/26/93        10:00            TAKEN
 06       01/26/93        10:30            TAKEN
 07       01/26/93        11:00            TAKEN
 08       01/26/93        11:30            AVAILABLE
 09       01/26/93        13:00            TAKEN
 10       01/26/93        13:30            TAKEN
 11       01/26/93        14:00            TAKEN
 12       01/26/93        14:30            AVAILABLE
 13       01/26/93        15:00            AVAILABLE
   Type slot number (2 digits) corresponding to Date & Time    [ __ ]
 PF1:HELP    PF3:QUIT     PF4:PROCESS SELECTED TIME                PF8:PG FWD
 You are not scheduled for any interview.
```

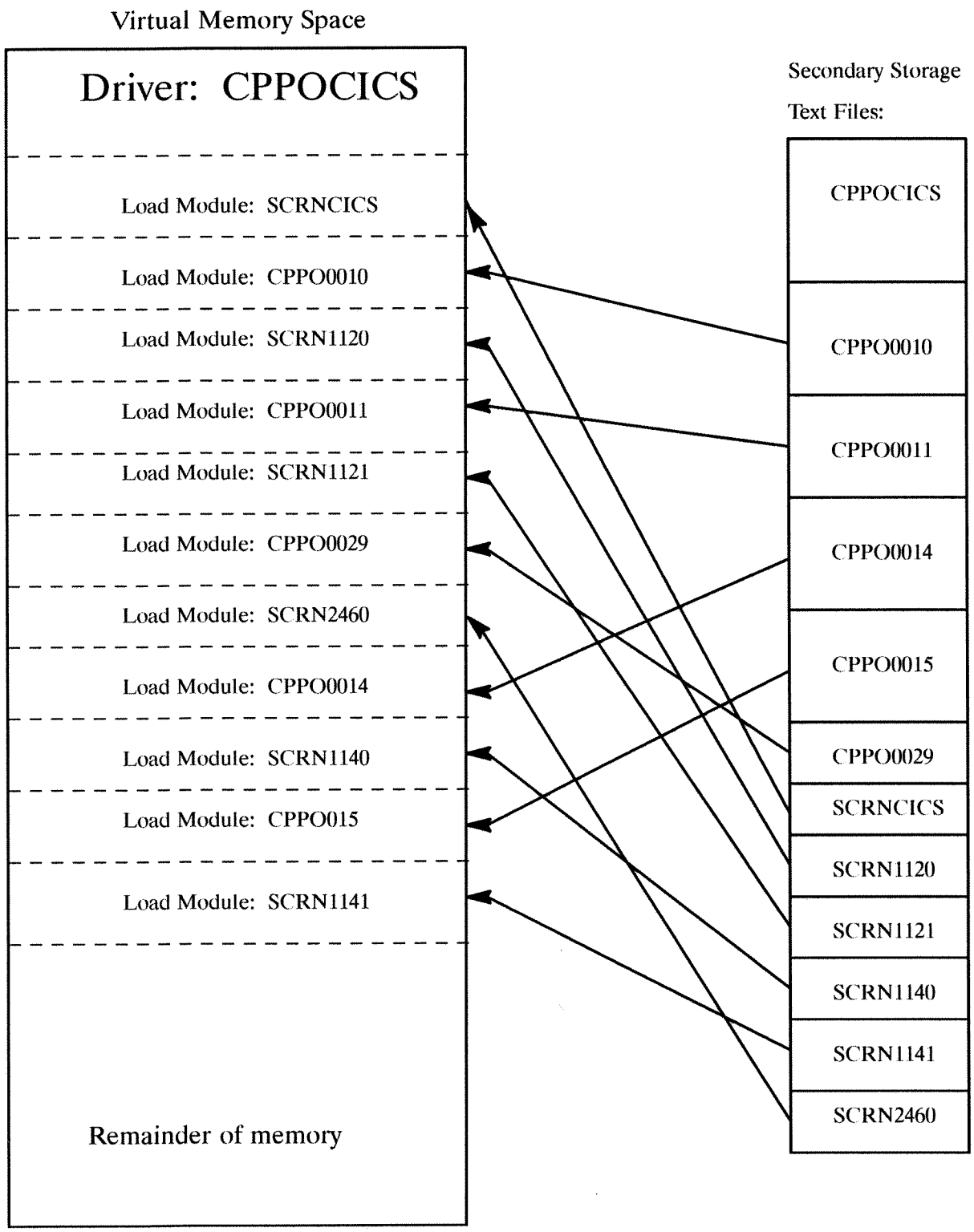
Figure 8.

Select Interview Screen.



When the submodules are dynamically compiled, they are called into memory at run-time and discarded when the submodules are complete. The link between the system program and the screen program is a static link.

Figure 9. Virtual Memory under Dynamic Linkage.



In a virtual environment, while at the CMS command line, the commands **LOAD CPPOCICS**, and **START** are entered. The test files that have been compiled in a STATIC fashion are loaded into memory. Each module remains resident in memory when not in use. When the **START** command is entered the driver begins to execute.

Figure 9.5 Virtual Memory under Static Linkage.

MINUTES FOR MEETING DATE: 10/1/92
Concerning the CPPO

Present: Joe Barry
Bill Miley
Steve Peter

The diskette version of the Placement Data Sheet (that is sold to employers) will require an additional program to read and write data to what may be a VM account. We could then use File Transfer to bring to a Personal Computer.

A hardware meeting is scheduled with the appropriate CPPO personel and Debbie Allision.

Concerning demographic data: Bill needs a list from me as to what information is needed to pull from the demographic data and the timing of it (how often; eg. twice a year). Keith needs permission from the Registrar to extract the data. He would submit a cover sheet and obtain Ken Bogard's permission.

Then there is the issue of when a student allows you to extract data then wants to revoke the permission. We also will need to account for the situation where a student enters demographic data, yet gives permission to extract data. Do we have the entered data wiped out?

Bill recommends that in the database, that all date fields have four digit years. Tables that include the same data should be given the same name.

We will have to include data for Minimum GPA. Because companies have the capability to state a "Prefered GPA." But in accounting they are allowed a minimum GPA.

The Plus/Minus grading system will not effect our database.

The database does not contain a personal identification or password. Will need an administrator to look-up passwords or to change passwords if the password cannot be read. How will it be handled, by phone call or personal visit. We also have to account for the initial logon; will the password have an initial value or will the user enter a value. If they enter a value they should enter it twice.

Bill asked if we wanted to our online system information regarding date/time/operator/terminal of last date of change. This could be used for student tracking for such claims that data was not entered in such a way.

Additional system requirements will include that of backup and recovery logic. Should backup occur once a day or after every transaction. A determination needs to be made on what data needs backup and when it should be performed (eg. before bid processing or before update scheduling). There most probably is a transaction log maintained by the system. Check with Steve Moore on this.

NOTES from phone call. DATE: 10/20/92
Concerning the setup of Multiuser Environment.

Parties: Steve Peter & Steve Moore

Steve Peter called Steve Moore to ask him to implement the third choice from our last meeting that uses a rotary time access to one of the VM accounts.

S. Moore said that he would introduce this to his manager. He said that he would probably need to get the principle people involved in a meeting that would cover options so that the investment of time is warrented. They would want to make sure that we are serious about the CPPO project. I assured him that we have made a commitment to the CPPO for this system. He said that if aproved (work to alter the operating system), that implementation would take just a few days.. There would be the examination of the operating system and the development of the system modifications. The modifications would have to be applied and tested.

October 12, 1992
Meeting with Steve Moore
to Investigate How Users
Will Access the CPPO System.
==========

The base level portion of the operating system is the Control Processor (CP). CMS is built on top of the CP. The CP allows jobs to have a time slice, and in a sence, allows more than one job to run at a time. This is done through the partitioning of memory into its virtual segments. This virtual concept applies also to disks and terminals.

CMS on the otherhand, will not allow a single user to run more than one application at one time. CMS allows users to access packages and edit files, run programs ect. Since CMS is built on CP, multiple users can be running the same application at the same time.

In addition, SQL allows multitasking; it has the capability to take care of locks ect. to maintain system. It can control concurrent access.

The Virtual Machine Connumication Facility passes messages from CMS to the CP which may go to various applications held active like SQL.

LOGON Procedures
----- ----------

We will alow users to logon, register with the system and allow them to access the context operation environment. They will not be allowed to escape to CMS where they could launch another application. The application security must prevent users from being able to communicate with CP. If they are able to IPL (which I can't see where they would in our system) then they must be sent right back into the CPPO application.

There are three means to implement the logon process.
1) The easiest means would be to have "N" accounts with userids ranging from CPPO0001 .. CPPOOOON. The advantage of this method is that it wouldn't require very much additional work from the techical support staff in implementation. The disadvantage to the user may be that they may have to attempt several userids until a free one is found.

2) The second would involve the dedication of terminals. Perhaps terminal could be set at Kreger, Library and the CPPO. Those terminals would already be logged-on to the CPPO System and would be active. The advantage of this plan would eliminate the logon conflict. The disadvantage of this approach is that if the system goes down, there would have to be an operator that would need to launch the application from the terminal.

3) The third technique would involve a Text Based Information Display. Here the user would know the name of the application and present it at the userid entry field. Behind the scene, the operating system would find an available Virtual Memory Segment. The advantage of this means would be that it would be easy for users to logon to the system. They could access from almost any terminal on campus or through modem. The disadvantage is that it would allow more work on our techical support people, and they would need to get approval to spend time toward the

project.

Two Notes in Conclusion:

There are two levels of security: SQL security & Application security.
Once the system is fully implemented who will maintain it?

MINUTES FOR MEETING DATE: 10/15/92
Concerning the CPPO

Present: Joe Barry
Bill Miley
Steve Peter

Bill Miley called attention to a new possibility concerning the adding of "EMPLOYER" data to the database. He said that it could be poosible for the CPPO personel to key in employer data as they do now on their WANG OAS system. The systems support could then on demand upload the new data.

Joe Barry is checking on the number of digits required in the employer number since there are some 2500 employers listed at the CPPO. Currently though the CPPO's system uses a three digit number.

Joe is also looking into how they want to present the GRAD_DATE on screen SCRN1121. The question is: what is easiest for the user. Is it easier to understand M/93 or 05/93. Discussion continued on the storage of this field in the database as a four digit year.

The new database design was presented to Bill and Joe.

The outcome of the meeting related to "Multiuser Access" with Steve Moore and Steve Peter was covered. {See meeting with Steve Moore} I need to contact Steve Moore to check the possibility of option three and set a time on its implementation.

There was a question in our last meeting that Joe Barry checked on. It concerned the situation where a student forgets their password. The solution requires the student to come to the CPPO and see the Administrator (which may be Rich Hearin). Bill recommended that there be two Administrators.

As for maintenance after implementation, the undertaking shall be handled by University Information Systems. A Request will have to be made for such services. Stearing groups will decide which system receive priority.

Bill will provide me with what he is would like for General Documentation.

The matter of the need for FORMS SOFTWARE to print the Placement Data Sheets was expressed by Steve Peter. Bill mentioned that some of the existing Lasers have a Forms Definition Language. One example of a document printed by FDL is the annual W2's. The University Information Systems has an "Ion Printer", and they print some items for the CPPO at this time. Perhaps the Placement Data Sheets could be printed at the UIS.

MINUTES FOR MEETING DATE: 10/26/92
Concerning the Multiple Access to the CPPO System.

Present: Steve Moore
John Kinney
Tom Schaber
Steve Peter

John Kinney felt as though it is important that we address the issue of multiple access properly so that problems can be eliminated, and that any inherant problems with the CPPO System do not become a relection on those as ACS.

He felt that there would be difficulty with the first time user logging on to the system via remote logon. Tom said that we will then consider the Remote logon a bonus (not a feature in our system).

It was mentioned that there would be a problem with the VM Rotary Dial Service. That problem requires a change to the Operatiing System which is something that John would like to avoid.

Instead he recommends issuing accounts to CPPO users. There would be about 3000 CPPO users.

There are problems with the 3000 account design.

1. Who will generate? CPPO? That means the CPPO must get involved with the management of these accounts. There would be two types of accounts in this design. Some of the 3000 will already have accounts and some will not. Those with accounts will use there same account and will not get a CPPO account. The others would get a CPPO account.

The CPPO would get a list of account names from ACS and they would note whether a given account goes directly into the CPPO System or whether they can go into the system from the account they already have from their course of study. Then there is the additional task to the CPPO to handle people who forget their password.

2. The CPPO system as it is designed now uses Static Calls. If there are 100 CPPO virtual users with static calls on the system, it could be a drain on the system.

Whether we go with Rotary Accounts or Individual Accounts there is a security issue involved. We must prevent users from having access to read or update the database from outside the CPPO system.

I can see how security could be important to the person who has an account from another course of study but for those CPPO accounts, I do not see how someone could break out of the CPPO System.

We are not sure whether it is possible to issue authority to change the database only through the programs and not by one's account. Do we Grant Authority Public to avoid the tedious issue to individual accounts? An investigation is needed.

MEETING TIME: 1.5 hr
MEETING TO CONTINUE: Weds. 10/28/92

MINUTES FOR MEETING DATE: 10/28/92
Concerning the Multiple Access to the CPPO System.

Present: Steve Moore
John Kinney
Tom Schaber
Steve Peter

Tom had good news. I was there to observe his check on database security. He created a module to access a database relation. He GRANTed authority of the relation to the program (and not to any students). The result is that the module runs and can access the relation but the student cannot when out of the program.
This means that the account that creates it can GRANT access to the modules. Therefore, no individual (except the DBA) will have authority to access those tables. The only way that the security of the database would be in jeopardy is if a person breaks into the DBA's account.

As for user access, Barb Edwards has (prior to our concerns) been working on a policy where all students will be issued accounts from the point were they enroll in the university to the point that they leave. This means that all CPPO applicants will already have accounts.

Another issue brought up was whether the current system design of Static calls will present a problem with memory space usage. The question was asked, how many lines of code are there? There are about 50 program modules and 50 screen program module each with 2000 lines of code. That's 200,000 lines of code. It was mentioned that I look into what it will take to utilize dynamic calls.

MEETING TIME: 25 minutes

MINUTES FOR MEETING DATE: 10/29/92
Biweekly Meeting

Present: Linda
Mary Alice
Joe Barry
Bill Miley
Steve Peter

Bill requests the specs on the STUDENT Table. I told him that the database has taken on changes (even this week). I think that after these changes, that it may now be appropriate to indicate the demographic data that will be needed from the registrar.

Bill asked how will it be determined what students get downloaded from the registrar. Shall that be all Seniors and Juniors or what. The CPPO will provide this information.

Bill mentioned that it would be nice if there was an additional menu option of the CPPO System from the MicroCenter computers.

Bill has asked for a schedule of events for the project. I told him that I am revising our database reloading facility now. In the next weeks I will be working to make our system work with dynamic calls (not static as it is now) and a study on Forms Software. I expressed that I thought students from SAN 475 would be taking an active role about this time. In a meeting this week, the system size was discribed as being about 50 modules. I have six modules completed which are related to checking a schedule, obtaining information on an interview, placing a bid, bid processing and selecting an interview slot.

Joe asked and stressed that the CPPO would like "teacher candidates" to have the capability of accessing the system via a branch campus. There shouldn't be a problem with this if the branch campus has access to the network.

Joe mentioned that students need the capability to change the GPA field. Need this option for transfer student.

Mary Alice pointed out that the system should be able to produce an EMAIL message to those students effected by a late cancelation of an employer. It was mentioned that a first attempt is made to call the students. All we would need is to provide the userid's of the students to were the email message should be sent.

Linda remarked that should be able interviews in which there are no bids, just a sign-up process. (eg. interviews for Peace Corps, or company that sets up an interview date to soon for the bidding process to occur.) Linda will be helping us by providing a list of exceptions as soon as she possibly can. (Exceptions like: handling Alumni that are not registered for closed interviews.)

Mary Alice stated that it would be nice if we included in the system a screen or portion of a screen that will list a "Job Discription." Not every position has a job discription. A job discription may be a paragraph or a full page.

The question was asked whether a company could issue the same password

to students interviewing for the same position? Designers will ponder this proposal. But the CPPO needs to come up with a policy to handle closed interview violations.

It was noted that our system could be used to handle "the Big Six Night."

NEXT MEETING IS: Nov. 12, 1992

MINUTES FOR MEETING DATE: 11/12/92
Biweekly Meeting

Present: Mary Alice Grassmick
Linda Smith
Joe Barry
Bill Miley
Steve Peter

It was asked whether a student will be able to check the VISIT schedule by date. It was also mentioned that it would be great if the list could be seen from both a date order and a alphbetic order in the system. I explained that the design is order by date.

Joe stated that he could get students for testing, maybe those students who have accepted offers or some other group. I said that would be an excellant idea. This type of testing is month away though. This type of testing may be useful in April or May.

Another thing needed with the GPA field is a flag. This data for this field will be originally obtained through the registrar. It would be nice to know when a student has changed this field. Otherwise we would need to compare this field in the database with the field in the original sequential file obtained from the registrar. (or add another field to database called ORIGINAL_GPA

Cut off times: A student can cancel an interview up to 9 A.M. the day prior to the interview. The lastest a student can sign-up for open sign-up slots is up to 1 P.M. the day before the interview. The deadline is needed to give preparation time for the next day's interviews. We need to make sure that when a student cancels an interview that another student cannot grab that slot that has become open. Because there is an Overflow list that should be contacted for those slots. This maintains the "fairness" in the system. Those that notify early and get on the overflow list should be the ones that get the opportunity in a last minute opening. If it were the "first see, first get" situation then we might have a problem of Fraternity/Sorority (or other groups) control (obtaining interviews for friends, canceling ect so they can get the interview).

In printing, we need to make sure that staff will be able to print a specific student's data sheet.

Clarifications to the "RESPONSE TO QUESTIONS" dated 11/11/92

#4. The student usually initiates the contact with the company in such a case. Students will be told by phone or letter. The student will then notify the CPPO. So we will monitor this loosely. We will take the students word for it. If the student is found to misrepresent the truth, then they will face a penalty from the CPPO. There will be a need to modify the number of slots that a interview may have. About 90% of the companies keep the slots as planned. Some companies add and some delete the number of slots. There was a case recently were a company had to move a few of its earlier interview slots to later in the day (after the previously scheduled last time slot). How can this be handled? We will have to add this feature to the system.

#5 Need overflow list. There are students on the overflow list that do not bid. Students must meet company requirements. There must be "print" capability. See "fairness" issue above.

#6 Staff need the ability to check the information that they have just entered. This can be through a view (by screen) or by printout.

#8 Cancel interview up to 9 A.M. the previous day.
Open sign-ups end at 1 P.M. the previous day.

#9 on SCRN2320, the field "Winners Must Schedule By" should be renamed to "Open Sign-ups begin."

#10 A correction was made to the stated group of student who will have logon capabiltiy. That group was stated as Juniors and Seniors. It actually is Seniors and Accounting Juniors. Others will have to contact the CPPO.

#11 Solution: The field in VISIT relation called ADDITIONAL_REQMTS will contain a note similar to this: "Company requests that you interview with only one of their divisions." There is a limit on the length of this field though. A situation arrises where a student places bids on the two different divisions and wins both but can only accept one interview. The way the system is designed, the student will spend points on the interview that they cannot take. They will have the see the CPPO if they care to recover these points.

Bill and Steve went through the Student Relation and noted those fields that can be obtained from the Registrar. Some of the fiels are questionable, so Bill is checking the Registrar's data.

NEXT MEETING IS: Dec. 3, 1992; 9 A.M.

Notes: 11/12/92

TO: Joe Barry
Mary Alice Grassmick
Linda Smith

From: Steve Peter

With the flag field that was presented in the meeting today, we need to know if this is something that will get a fair amount of use to warrant its addition to the system. How often would it be needed? Without it in our system, someone like Bill Miley could be contacted and asked if the student has changed the GPA field. That person probably give you an answer back probably by the next day. All that would be involved is comparing the GPA that is in the database with the original sequential file that was obtained from the Registrar. I'll send Bill a Note to see if this is a task that his department would like to avoid. And tell us if you would like the system to provide you with that information. If so, how would you like the information to be provided to you? By screen or report.

In regard to CUT OFF TIMES, when a company has three days of interviews for the same position, will we have three different cut off times or just one based on the first day? From the development side, it would be easiest if the cut off times were to based on the day of the interview. Is this the best for you?

On the topic of the Overflow list, this is not in our system at this time. When the issue came up we thought that we could use the students who place bids as a guage of interest. If you think about the system as it is now, and ask yourself why somethings are the way they are, you may conclude that the system has evolved to meet the problems associated with an inability to judge student interest at the last moments while keeping the system "fair". Maybe you do not agree but lets see. It seems like you still would like to call the students so that you are sure to fill the slots. Is that correct? So you need a list. What would be the difference between the overflow list as it is now and getting a list based on the bid losers? This list could be presented in order by points bid, or by date and time of bid. If this list would be based on order by date and time, it would be very similar to the overflow list with the exception that the student did not have to put the extra effort into getting on the list as they would under the overflow method. We did discuss that there are those who go on overflow that do not bid. But with the loser list method, students will have to plan ahead and make sure that they place a bid on the company (putting in effort and showing interest in the company). With the overflow method, how "interested" is a student in a company if they do not place a bid? Of course there is the situation where a student is down to their last couple hundred points or have no points left. How can they become overflow students if the list is based on bid losers ranked by bid points? True, but such a student has shown alot of interest in many other companies, and they have apparently had some use of the CPPO facility (more than most students). How fair is that to the student who has hardly used the facility, has plenty of points and meets the company requirements? This is quite interesting. There are many

ways to design the new system.

In conclusion, the options are:

institute an overflow list into the system,
use the losing bidders ranked by date and time,
use the losing bidders ranked by amount of bid,
or have a system where there are no "open sign-ups" and use the next bid loser with the next highest bid amount for the open sign-up, then the ones left after that can be used for overflow, ect.

The advantage of using the losing bidders for overflow is that we would be minimizing the amount of data produced and needed by the system. It would be more efficient. We would be using the minimum amount of data required to run the system.

The disadvantage is that it is different from the way the system is now. Tell me what you think. Feel free to leave a message on my recorder or send me a EMAIL message. My userid is SPETER. Thank you.

November 11, 1992
Response to Questions
==========

1) WILL ONLY THOSE STUDENTS WHO MEET THE REQUIREMENTS OF AN ORGANIZATION BE ABLE TO GET ON AN OPEN SCHEDULE?
Yes, those students that meet the requirements of certain companies will see those companies on Screen 1.1.2. [See Figure 1]

2) WILL WE BE ABLE TO CONTROL WHEN SIGN-UPS STOP FOR A PARTICULAR SCHEDULE?
Yes, see Figure 2. This screen can only be accessed by CPPO personel. The last two entry fields will control the last sign-up date. Once this is entered, we have not planned for a change in midstream.

3) WILL WE BE ABLE TO CONTROL WHEN A SCHEDULE GOES OPEN? Yes see Figure 2. After the date entered in the third last input line, the system will allow the interview to go open if slots are available.

4) MANY TIMES A STUDENT IS TOLD BY THE ORGANIZATION THEY CAN BE ON A SCHEDULE WHEN THEY DO NOT MEET THE REQUIREMENTS. HOW WILL THEY BE ABLE TO GET ON THAT SCHEDULE. We have not worked this into the system. Before we add it to the system I'll need to know more about the following. How are the students going to be told? By letter or phone call? How closely do you want to monitor this? We could probably work it in similiar fashion as the closed interview.

5) WHEN A SCHEDULE IS FILLED, HOW CAN STUDENTS SHOW THEIR INTEREST IN THE ORGANIZATION? (WE NOW HAVE OVERFLOW LISTS). WILL DATA SHEETS OF STUDENTS ON OVERFLOW ALSO BE PRINTED OUT. ALSO, WILL THE STUDENT BE REQUIRED TO MEET THE ORGANIZATION'S SPECIFICATION? In our early design we proposed printing all students who bid on an interview. But now we have planned to print the Placement Data Sheets needed for interviews. We can make a modification here. Would you like it if we printed the top certain percent of bidders? What should that percent be? Or should we have every bidder for an interview printed?

6) WE NOW HAVE A RECOVERY DECK TO USE IF AN ERROR HAS BEEN MADE WHEN ENTERING INFORMATION. WILL THERE BE SOME CHECK/BALANCE TO BE SURE INFORMTIONIS ENTERED CORRECTLY. Yes, there is an error check on entry fields except for student entry fields related to the Placement Data Sheet.

7) WILL STUDENTS BE ABLE TO VIEW A LISTING OF ALL ORGANIZATIONS THAT ARE SCHEDULED OR WILL THEY BE ABLE TO SEE ONLY THOSE SCHEDULES FOR WHICH THEY ARE QUALIFIED? [See Figure 4] Yes, option four will provide that service.

8) IF A STUDENT WISHES TO CANCEL AN INTERVIEW, HOW WILL IT BE HANDLED? ALSO, DURING WHAT TIME FRAME MAY AN INDIVIDUAL CANCEL AN INTERVIEW?
I didn't get a screen print for this one but there is an option for a student to cancel an interview. But to my knowledge, we have not placed a time limit on the cancel feature. What would you like?

9) OCCASIONALLY, BECAUSE OF A SHORTAGE OF TIME, AN ORGANIZATION WILL SET UP A SCHEDULE AND IT WILL BE FOR OPEN SIGN-UP ONLY. WILL SIGN-UPS BEGIN AS SOON AS IT IS ENTERED IN THE SYSTEM, OR CAN A DATE BE SET FOR OPEN SIGN-UP TO BEGIN? Yes in Figure 2, to have an interview become

initially open, you will have to enter "today's date" in the last two input lines: "Last Day to Schedule" and "Winners Must Schedule By."

10) HOW WILL INTERN, OTHER THAN ACCOUNTING, SCHEDULES BE HANDLED? WILL THE SIGN-UPS BE DONE MANUALLY IN THE CPPO? ALSO, INCLUDED IN THIS GROUP WOULD BE THE PEACE CORPS, BOY SCOUTS OF AMERICA, UNIVERSITIES, ECT. Is it OK if we use the same employer number (for that given company) and a unique schedule number? We plan on giving all Seniors and just Junior Accounting and System Analysis majors the logon capability (a change recommended by Dr. Schaber). They will have to enter the data for the Placement Data Sheet. Interns less than Junior will require a physical addition by the CPPO staff. The addition will involve adding a social security number and name.

11) WE NOW HAVE COMPANIES COMING WITH SEVERAL DIVISIONS AND ASK THAT STUDENTS INTERVIEW WITH ONLY ONE DIVISION. HOW CAN WE CONTROL STUDENTS PLACING BIDS OR "OPEN SIGN-UP" OF MORE THAN ONE DIVISION. FOR EXAMPLE BANK ONE OR NCR. If there are two different schedules there is no easy way to implement this at this point. This is a substantual modification, and work on this will delay implementation. Perhaps we can place in the next version.

MINUTES FOR MEETING DATE: 11/23/92
Concerning the purchase of forms software and laser printer.
Present: Rich Harin
Joe Barry
Steve Peter
Linda Smith

Went over the procedure for purchasing forms. Steve took "Requisition Sheet" to fill out.

Found out that the CPPO has just received a HP Laser 4M.

Rich wants to know how students can print their own data sheet. I explained to him that the lab is equiped with "pay for print" lasers. The only problem is that these lasers will not possess the forms software that will have the form image. How will students obtain extra data sheets to distribute to employers?

Well, they really won't have a need to print the form. A student may want to print the output that would be on the data sheet so that they can verify it. We will make sure that they have that capability. As for company distribution, students can use their own personal resumes.

Linda Smith has presented the following questions:
WE HAVE REQUESTS OCCASIONALLY FROM COMPANIES RECRUITING FOR MASTERS AND BACHELORS (EACH HAVING A DIFFERENT SCHEDULE). IF ONE FILLS UP AND THE OTHER DOES NOT, WILL IT BE POSSIBLE TO HAVE THE OVERFLOW FROM ONE GO INTO THE OTHER.

We really haven't determined how we are going to handle overflow in the first place. There are many ways that we can handle overflow as covered in our last coorespondence, and the overflow procedure is usually handled manually. The only question is "how will the overflow list be determined?" I mentioned this to Rick and Joe moments earlier. It is something that will be determined.

WITH THE SAME IDEA IN MIND, WILL WE BE ABLE TO APPLY THE SAME CONCEPT TO THOSE INTERVIEWING FOR INTERNSHIPS AND FULL-TIME POSITIONS? Same answer as above.

WILL INTERDISCIPLINARY MAJORS BE CODED FOR BIDDING WITH THEIR "AREA OF FOCUS?" We'll have to put this in the system. First the student will have as their major "interdisciplinary studies", then we need to make sure that they have an area for them to indicate their "area of interest." Then the CPPO personel or the student themselves will enter a three digit code related to this area of interest. These codes are on a form that I obtained in the meeting.

WILL MBA'S BE ABLE TO BID, OR SIGN-UP THROUGH "OPEN" SIGN-UP USING THEIR UNDERGRADUATATE DEGREE (AS THEY CAN NOW)? We will have to make sure that they can.

MINUTES FOR MEETING DATE: 12/03/92
Biweekly Meeting

Present:

Mary Alice Grassmick
Linda Smith
Joe Barry
Bill Miley
Steve Peter

Discussed overflow. Determined that there would be no overflow list.
a late cancelation will leave a slot open but
"that happens anyway."

Program CPPO0015 should now allow a cancelation after all slots are filled because it will go back to open sign-up.

Will company be able to change things in midstream sonce some students have bid. For example: open bid to all. To open to May graduates.
to allow half masters, half interns.
There are just 2 or 3 companies tht cause exceptions

Bill Miley gave me the registrar's major codes which are two digit numeric. The CPPO uses there own 3 character abreviation.

NEXT MEETING: DECEMBER 17, 1992.

MINUTES FOR MEETING DATE: 12/17/92
Biweekly Meeting

Present: Rich Hearin
Mary Alice Grassmick
Linda Smith
Joe Barry
Bill Miley
Steve Peter

Discussed my project progress.

The question came up: Is there another graduate student going to take over the project?

Issue of codes for major was brought up. Codes for register are 2 digits numeric. CPPO uses an abreviation for the major.
eg. TEN for english teacher
TSX for secondary certified teacher
EGR is shown by the CPPO as MFE.
and when students enter, we need to validate their entry.
Maybe we should have the student just enter and not worry about the register information on major.
We will then have to display to student a list prior to entry.
We could have student enter a numeric code and then transfer that to a three character code.

Rich felt that there may be a need for staff to get in and change student major. For example. MBS. a masters in business. They should be allowed to interview in their undergraduate degree. It was mentioned that maybe there could be a question to the user: What was your undergraduate major?

How are students going to get there points in the beginning? Expected grad date will be obtained through registrar. The student will be able to change this info.

Graduates get:

December	7500 pts.
May	10000 pts,
Education	10000 pts.
Accounting Interns	5000 pts, may have to

consider GPA for accounting interns at a future date.
Then there are those situations where a student graduates in May and takes classes in the Fall. Staff will have to be able to award points.

Note: need to be able to restrict Junior graduates from getting those companies that they are not eligible to bid on.

The staff need access to info on the computer information form and the Student Placement Data Sheet.

The question was brought up: Do we have enough terminals to support thi: system? There are about 2500 students. Is there enough hardward available so that students can register and input data? "What they don't need is a system that works, but people can't get to."

There is a survey that is given to intership applicants. (See attached

form) The information it includes is: student name, phone number, major, name of organization, location, name of official to contact for information, company address, company phone, type of interhip, describe duties (6 lines), a question (Was internship paid or non-paid?, salary. This information would not need to be in a perminent database. It could if we want it to be. But staff said that they would be happy to print and file. They would like to be able to know how many students had interships. Reason: To let other students know where the interships were.

THIS IS THE FINAL MEETING FOR STEVE PETER.

Miami University
CAREER PLANNING AND PLACEMENT OFFICE
228 Hoyt Hall
Oxford, Ohio 45056

INTERNSHIP SURVEY

If you have had an internship, please complete this internship survey form and return it to the Career Planning and Placement Office when you submit your registration materials. We would appreciate it if you would list any internships you may have had, regardless of the type of organization. Additional forms are available in the Career Planning and Placement Office. This information is kept in our Career Resource Center and used as a resource for other Miami University students looking for internships. We appreciate your cooperation very much.

Please Print

Your name _____

Phone (_____) _____

Major _____

Name of organization _____

Location _____

Name of official to
contact for information _____

Address _____

_____

_____

Phone (if known) (_____) _____

Type of internship _____

Describe duties _____

_____

_____

_____

_____

_____

Was internship paid or non-paid? _____

Salary _____

b. Figures

BEGINNING OF FIGURE 10.

```
/* EXEC TO PRECOMPILE, COMPILE AND TEST A COBOL PROGRAM          */
address command
GLOBAL TXTLIB VSC2LTXT DFSRTLIB SQL XMENUSUB CMSLIB
GLOBAL LOADLIB VSC2LOAD
'SET LANGUAGE AMENG (ADD ARI USER'
ARG filename
'EXEC LOGEDIT1'
XEDIT filename COBSQL
'EXEC LOGEDIT2' filename
'PIPE',
'CMS SQLPREP COB PP(PREP='filename',APOST) IN ('filename' COBSQL A) |',
'> DUMMY FILE A'
if rc > 0 then
  do
    sql_rc = rc
    'EXEC LOGPREP1'
    say 'SQLPREP did not complete successfully.'
    if sql_rc = 4 then do
       say 'SQL WARNINGS occured which must be investigated.'
       say 'You will be placed in XEDIT so you can locate the error.'
       say 'Find where the warning occured by searching for the word "WARNING".'
       say 'After determining the cause of your problem, you will have to'
       say 'restart SQLCOBOL' filename'.'
       say ' '
       say 'Press any key to begin the review of the listing.'
       pull RESPONSE
       PUSH 'LOCATE /WARNING/'
       'XEDIT' filename 'LISTPREP A'
       'EXEC LOGPREP2' filename
    end
    if sql_rc > 4 then do
       say 'SQL ERRORS occured which must be corrected.'
       say 'You will be placed in XEDIT so you can locate the error.'
       say 'Find where the warning occured by searching for the words "SQL ERROR".'
       say 'After determining the cause of your problem, you will have to'
       say 'restart SQLCOBOL' filename'.'
       say ' '
       say 'Press any key to begin the review of the listing.'
       pull RESPONSE
       PUSH 'LOCATE /SQL ERROR/'
       'XEDIT' filename 'LISTPREP A'
       'EXEC LOGPREP3' filename
    end
    exit
  end
else
  do
    'ERASE' filename 'LISTPREP A'
  end
'EXEC LOGCMPL1'
'COBOL2' filename '(APOST SIZE(2000K) TEST XREF DYNAM RES FLAG(W,W) LIB   SSR'          <====== THIS IS LINE 51.
COMPILE_RC = RC
'EXEC LOGCMPL2' filename
IF COMPILE_RC > 5 THEN
   DO
     SAY 'SYNTAX ERRORS IN COBOL SOURCE PREVENT EXECUTION.'
     SAY 'FIX AND RESUBMIT.'
     'EXEC LOGEASY1'
     'EXEC EASYFIND' filename
     'EXEC LOGEASY2' filename
     EXIT
   END
```

```
FILEDEF SYSABOUT DUMMY
/*  remove debugger for initial use of sqlcobol
'EXEC LOGDBUG1'
'EXEC ISPF 2'
'EXEC LOGDBUG2' filename
'ERASE filename COBOL'
*/
```

END OF FIGURE 10.

The EXEC that Initiates the Compilation.

BEGINNING OF FIGURE 11.

```
Ready; T=0.01/0.01 12:33:00
STORMAP
                                Storage Map
                                ------- ---

VMSIZE          NUCALPHA        NUCSIGMA        NUCOMEGA
00400000        00E00000        00FEDB98        01000000

                                Unallocated Free Storage Queue
                                ----------- ---- ------- -----

      <16Mb                             >16Mb
Total           Largest         Total           Largest         Total Unallocated
0030B000        002F1000        00000000        00000000        0030B000
Ready; T=0.01/0.02 12:33:06
SPOOL CONS STOP CLOSE
```

END OF FIGURE 11. Storage Map Display.

BEGINNING OF FIGURE 12.

```
21:32:46 OBTAINED BYTES=000000C8 ADDR=003E8428 SUBPL=USER      CALLER=000240FC
21:32:46 OBTAINED BYTES=000000B0 ADDR=003E8798 SUBPL=USER      CALLER=000241F0
21:32:46 OBTAINED BYTES=00000798 ADDR=003680D8 SUBPL=USER      CALLER=00024358
21:32:46 OBTAINED BYTES=000002C0 ADDR=003E8D00 SUBPL=USER      CALLER=000243E4
21:32:48 OBTAINED BYTES=000000C8 ADDR=003E8428 SUBPL=USER      CALLER=000240FC
21:32:48 OBTAINED BYTES=000000B0 ADDR=003E8C50 SUBPL=USER      CALLER=000241F0
21:32:48 OBTAINED BYTES=000007B0 ADDR=003A1850 SUBPL=USER      CALLER=00024358
21:32:48 OBTAINED BYTES=000002C8 ADDR=002C1D38 SUBPL=USER      CALLER=000243E4
21:32:58 OBTAINED BYTES=000000C8 ADDR=002C1C70 SUBPL=USER      CALLER=000246E0
21:32:58 OBTAINED BYTES=000000C8 ADDR=002C1F38 SUBPL=USER      CALLER=000240FC
21:32:58 OBTAINED BYTES=000000B0 ADDR=003E8C50 SUBPL=USER      CALLER=000241F0
21:32:58 OBTAINED BYTES=000007C0 ADDR=00365840 SUBPL=USER      CALLER=00024358
21:32:58 OBTAINED BYTES=000002C8 ADDR=002C1C70 SUBPL=USER      CALLER=000243E4
21:33:06 OBTAINED BYTES=000000C8 ADDR=002C1BA8 SUBPL=USER      CALLER=000246E0
21:33:06 OBTAINED BYTES=000000C8 ADDR=002C1F38 SUBPL=USER      CALLER=000240FC
21:33:06 OBTAINED BYTES=000000B0 ADDR=003E8C50 SUBPL=USER      CALLER=000241F0
21:33:06 OBTAINED BYTES=000007B0 ADDR=00365850 SUBPL=USER      CALLER=00024358
21:33:06 OBTAINED BYTES=000002C8 ADDR=002C1C70 SUBPL=USER      CALLER=000243E4
21:33:11 OBTAINED BYTES=000000C8 ADDR=002C1BA8 SUBPL=USER      CALLER=000246E0
21:33:13 OBTAINED BYTES=00000050 ADDR=003E8B08 SUBPL=USER      CALLER=00024868
21:33:13 RELEASED BYTES=00002010 ADDR=00366000 SUBPL=USER      CALLER=000249C4
21:33:13 RELEASED BYTES=00000108 ADDR=002D0418 SUBPL=USER      CALLER=0000E286
```

END OF FIGURE 12. Static Memory Usuage.

BEGINNING OF FIGURE 13.

```
16:41:01 OBTAINED BYTES=00000400 ADDR=003CE000 SUBPL=DMSOS000 CALLER=00028206
16:41:01 OBTAINED BYTES=00000338 ADDR=003CECC8 SUBPL=DMSOS000 CALLER=003E6AD8
16:41:02 OBTAINED BYTES=00002000 ADDR=003C9000 SUBPL=DMSOS000 CALLER=0039A276
16:41:02 OBTAINED BYTES=000000C8 ADDR=003B5230 SUBPL=USER     CALLER=000240BC
16:41:02 OBTAINED BYTES=00002010 ADDR=003A1000 SUBPL=USER     CALLER=0002A078
16:41:02 OBTAINED BYTES=00000050 ADDR=003B51E0 SUBPL=USER     CALLER=00028534
16:41:02 RELEASED BYTES=00000050 ADDR=003B51E0 SUBPL=USER     CALLER=000286D0
16:41:02 OBTAINED BYTES=000000B0 ADDR=003B5180 SUBPL=USER     CALLER=000241B0
16:41:02 OBTAINED BYTES=000000A8 ADDR=003B50D8 SUBPL=USER     CALLER=0002B25C
16:41:02 OBTAINED BYTES=00000850 ADDR=003A37B0 SUBPL=USER     CALLER=0002B3F0
16:41:02 OBTAINED BYTES=00000790 ADDR=003A3020 SUBPL=USER     CALLER=0002B610
16:41:02 RELEASED BYTES=000000A8 ADDR=003B50D8 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=00000798 ADDR=003E8828 SUBPL=USER     CALLER=00024318
16:41:02 OBTAINED BYTES=000002C0 ADDR=002D05C0 SUBPL=USER     CALLER=000243A4
16:41:02 RELEASED BYTES=000000C8 ADDR=003B5230 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002BCD0
16:41:02 OBTAINED BYTES=00000050 ADDR=003B50E0 SUBPL=USER     CALLER=0002DBE8
16:41:02 OBTAINED BYTES=00000128 ADDR=002D00D8 SUBPL=USER     CALLER=00031ED4
16:41:02 RELEASED BYTES=00000050 ADDR=003B50E0 SUBPL=USER     CALLER=0002E0F8
16:41:02 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=0002C028
16:41:02 RELEASED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=0002C028
16:41:02 RELEASED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=0002C028
16:41:02 RELEASED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=0002C028
16:41:02 RELEASED BYTES=000000A0 ADDR=003B50E0 SUBPL=USER     CALLER=00029EE0
16:41:02 OBTAINED BYTES=00000098 ADDR=003B50E8 SUBPL=USER     CALLER=0002CA78
16:41:02 OBTAINED BYTES=00000058 ADDR=003B5090 SUBPL=USER     CALLER=0002F114
16:41:02 RELEASED BYTES=00000058 ADDR=003B5090 SUBPL=USER     CALLER=0002F324
16:41:02 OBTAINED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002D8F4
16:41:02 OBTAINED BYTES=00000058 ADDR=003B5040 SUBPL=USER     CALLER=00031864
16:41:02 OBTAINED BYTES=00000790 ADDR=003E8098 SUBPL=USER     CALLER=000318DC
16:41:02 RELEASED BYTES=00000058 ADDR=003B5040 SUBPL=USER     CALLER=00031B48
16:41:02 OBTAINED BYTES=00000128 ADDR=002D00D8 SUBPL=USER     CALLER=00031ED4
16:41:02 OBTAINED BYTES=000000E0 ADDR=003B5518 SUBPL=USER     CALLER=000327C4
16:41:08 RELEASED BYTES=00000028 ADDR=003B5070 SUBPL=USER     CALLER=00032CB8
16:41:08 OBTAINED BYTES=00000128 ADDR=002D00D8 SUBPL=USER     CALLER=00031ED4
16:41:08 RELEASED BYTES=00000790 ADDR=003E8098 SUBPL=USER     CALLER=0002DFBC
16:41:08 RELEASED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002E0F8
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002F5FC
16:41:08 RELEASED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002F6D4
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002F9A4
16:41:08 RELEASED BYTES=00000050 ADDR=003B5098 SUBPL=USER     CALLER=0002FE74
16:41:08 RELEASED BYTES=00000098 ADDR=003B50E8 SUBPL=USER     CALLER=00029EE0
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002BE7C
16:41:08 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002BE7C
16:41:08 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002BE7C
16:41:08 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002BE7C
16:41:08 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:08 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=0002D31C
16:41:08 OBTAINED BYTES=00000050 ADDR=003B50E0 SUBPL=USER     CALLER=0002F9A4
16:41:08 RELEASED BYTES=00000050 ADDR=003B50E0 SUBPL=USER     CALLER=0002FE74
16:41:08 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER     CALLER=00029EE0
16:41:09 OBTAINED BYTES=000000C8 ADDR=003B5530 SUBPL=USER     CALLER=000240BC
16:41:09 OBTAINED BYTES=00000050 ADDR=003B56B0 SUBPL=USER     CALLER=00028534
16:41:09 RELEASED BYTES=00000050 ADDR=003B56B0 SUBPL=USER     CALLER=000286D0
16:41:09 OBTAINED BYTES=000000B0 ADDR=003B5650 SUBPL=USER     CALLER=000241B0
16:41:09 OBTAINED BYTES=000000A8 ADDR=002D0158 SUBPL=USER     CALLER=0002B25C
```

```
16:41:09 OBTAINED BYTES=00000870 ADDR=002F3040 SUBPL=USER     CALLER=0002B3F0
16:41:09 OBTAINED BYTES=000007B0 ADDR=003E8078 SUBPL=USER     CALLER=0002B610
16:41:09 RELEASED BYTES=000000A8 ADDR=002D0158 SUBPL=USER     CALLER=00029EE0
16:41:09 OBTAINED BYTES=000007B0 ADDR=002C1850 SUBPL=USER     CALLER=00024318
16:41:09 OBTAINED BYTES=000002C8 ADDR=002D02F8 SUBPL=USER     CALLER=000243A4
16:41:09 RELEASED BYTES=000000C8 ADDR=003B5530 SUBPL=USER     CALLER=00029EE0
16:41:10 OBTAINED BYTES=00000050 ADDR=002D01B0 SUBPL=USER     CALLER=000C0104
16:41:10 RELEASED BYTES=00000050 ADDR=002D01B0 SUBPL=USER     CALLER=000C03E0
16:41:15 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BCD0
16:41:15 OBTAINED BYTES=00000050 ADDR=002D0090 SUBPL=USER     CALLER=0002DBE8
16:41:15 OBTAINED BYTES=00000128 ADDR=003CDDD0 SUBPL=USER     CALLER=00031ED4
16:41:15 RELEASED BYTES=00000050 ADDR=002D0090 SUBPL=USER     CALLER=0002E0F8
16:41:15 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=0002C028
16:41:15 RELEASED BYTES=000000A0 ADDR=002D0090 SUBPL=USER     CALLER=00029EE0
16:41:15 OBTAINED BYTES=00000098 ADDR=002D0098 SUBPL=USER     CALLER=0002CA78
16:41:15 OBTAINED BYTES=00000058 ADDR=002D0040 SUBPL=USER     CALLER=0002F114
16:41:15 RELEASED BYTES=00000056 ADDR=002D0040 SUBPL=USER     CALLER=0002F324
16:41:15 OBTAINED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002D8F4
16:41:15 OBTAINED BYTES=00000058 ADDR=003E8020 SUBPL=USER     CALLER=00031864
16:41:15 OBTAINED BYTES=000007B0 ADDR=00304850 SUBPL=USER     CALLER=000318DC
16:41:15 RELEASED BYTES=00000058 ADDR=003E8020 SUBPL=USER     CALLER=00031B48
16:41:15 OBTAINED BYTES=00000128 ADDR=003CDDD0 SUBPL=USER     CALLER=00031ED4
16:41:15 OBTAINED BYTES=000000E0 ADDR=003CDE18 SUBPL=USER     CALLER=000327C4
16:41:17 RELEASED BYTES=00000028 ADDR=003B5240 SUBPL=USER     CALLER=00032CB8
16:41:17 OBTAINED BYTES=00000128 ADDR=003CDDD0 SUBPL=USER     CALLER=00031ED4
16:41:17 RELEASED BYTES=000007B0 ADDR=00304850 SUBPL=USER     CALLER=0002DFBC
16:41:17 RELEASED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002E0F8
16:41:17 OBTAINED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002F5FC
16:41:17 RELEASED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002F6D4
16:41:17 OBTAINED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002F9A4
16:41:17 RELEASED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002FE74
16:41:17 OBTAINED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002FD1C
16:41:17 RELEASED BYTES=00000050 ADDR=002D0048 SUBPL=USER     CALLER=0002FE74
```

```
16:41:17 RELEASED BYTES=00000098 ADDR=002D0098 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002BE7C
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=0002D31C
16:41:17 OBTAINED BYTES=00000050 ADDR=002D0090 SUBPL=USER     CALLER=0002F9A4
16:41:17 RELEASED BYTES=00000050 ADDR=002D0090 SUBPL=USER     CALLER=0002FE74
16:41:17 RELEASED BYTES=00000050 ADDR=002D00E0 SUBPL=USER     CALLER=00029EE0
16:41:17 OBTAINED BYTES=000000C8 ADDR=003CDE30 SUBPL=USER     CALLER=000246A0
16:41:17 RELEASED BYTES=000002C8 ADDR=002D02F8 SUBPL=USER     CALLER=0002A534
16:41:17 RELEASED BYTES=000007B0 ADDR=002C1850 SUBPL=USER     CALLER=0002A58C
16:41:17 RELEASED BYTES=000007B0 ADDR=003E8078 SUBPL=USER     CALLER=0002A640
16:41:17 RELEASED BYTES=00000870 ADDR=002F3040 SUBPL=USER     CALLER=0002A698
16:41:17 RELEASED BYTES=000000B0 ADDR=003B5650 SUBPL=USER     CALLER=0002A768
16:41:17 RELEASED BYTES=000000C8 ADDR=003CDE30 SUBPL=USER     CALLER=00029EE0
16:41:18 OBTAINED BYTES=000000C8 ADDR=003B5638 SUBPL=USER     CALLER=000240BC
16:41:18 OBTAINED BYTES=00000050 ADDR=002D0060 SUBPL=USER     CALLER=00028534
16:41:18 RELEASED BYTES=00000050 ADDR=002D0060 SUBPL=USER     CALLER=000286D0
16:41:18 OBTAINED BYTES=000000B0 ADDR=002D0000 SUBPL=USER     CALLER=000241B0
16:41:18 OBTAINED BYTES=000000A8 ADDR=002D0518 SUBPL=USER     CALLER=0002B25C
16:41:18 OBTAINED BYTES=00000878 ADDR=00304788 SUBPL=USER     CALLER=0002B3F0
16:41:18 OBTAINED BYTES=000007B8 ADDR=003E8070 SUBPL=USER     CALLER=0002B610
16:41:18 RELEASED BYTES=000000A8 ADDR=002D0518 SUBPL=USER     CALLER=00029EE0
16:41:18 OBTAINED BYTES=000007C0 ADDR=002C1840 SUBPL=USER     CALLER=00024318
16:41:18 OBTAINED BYTES=000002C8 ADDR=002D02F8 SUBPL=USER     CALLER=000243A4
16:41:18 RELEASED BYTES=000000C8 ADDR=003B5638 SUBPL=USER     CALLER=00029EE0
16:41:18 OBTAINED BYTES=00000050 ADDR=003B56B0 SUBPL=USER     CALLER=000C0104
16:41:18 RELEASED BYTES=00000050 ADDR=003B56B0 SUBPL=USER     CALLER=000C03E0
16:41:20 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002BCD0
16:41:20 OBTAINED BYTES=00000050 ADDR=003E8020 SUBPL=USER     CALLER=0002DBE8
16:41:20 OBTAINED BYTES=00000128 ADDR=002F3788 SUBPL=USER     CALLER=00031ED4
16:41:20 RELEASED BYTES=00000050 ADDR=003E8020 SUBPL=USER     CALLER=0002E0F8
16:41:20 PELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
```

```
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=0002C028
16:41:20 RELEASED BYTES=000000A0 ADDR=002C17A0 SUBPL=USER     CALLER=00029EE0
16:41:20 OBTAINED BYTES=00000098 ADDR=002C17A8 SUBPL=USER     CALLER=0002CA78
16:41:20 OBTAINED BYTES=00000058 ADDR=003B5628 SUBPL=USER     CALLER=0002F114
16:41:20 RELEASED BYTES=00000058 ADDR=003B5628 SUBPL=USER     CALLER=0002F324
16:41:20 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002D8F4
16:41:20 OBTAINED BYTES=00000058 ADDR=003E8018 SUBPL=USER     CALLER=00031864
16:41:20 OBTAINED BYTES=000007B8 ADDR=002F30F8 SUBPL=USER     CALLER=000318DC
16:41:20 RELEASED BYTES=00000058 ADDR=003E8018 SUBPL=USER     CALLER=00031B48
16:41:20 OBTAINED BYTES=00000128 ADDR=003CDDD0 SUBPL=USER     CALLER=00031ED4
16:41:20 OBTAINED BYTES=000000E0 ADDR=002F3018 SUBPL=USER     CALLER=000327C4
16:41:25 RELEASED BYTES=00000028 ADDR=003B5240 SUBPL=USER     CALLER=00032CB8
16:41:25 OBTAINED BYTES=00000128 ADDR=003CDDD0 SUBPL=USER     CALLER=00031ED4
16:41:25 RELEASED BYTES=000007B8 ADDR=002F30F8 SUBPL=USER     CALLER=0002DFBC
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002E0F8
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002F5FC
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002F6D4
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002F9A4
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002FE74
16:41:25 RELEASED BYTES=00000098 ADDR=002C17A8 SUBPL=USER     CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER     CALLER=00029EE0
```

```
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002D31C
16:41:25 OBTAINED BYTES=00000050 ADDR=003E8020 SUBPL=USER      CALLER=0002F9A4
16:41:25 RELEASED BYTES=00000050 ADDR=003E8020 SUBPL=USER      CALLER=0002FE74
16:41:25 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:25 OBTAINED BYTES=000000C8 ADDR=002F37E8 SUBPL=USER      CALLER=000246A0
16:41:25 RELEASED BYTES=000002C8 ADDR=002D02F8 SUBPL=USER      CALLER=0002A534
16:41:25 RELEASED BYTES=000007C0 ADDR=002C1840 SUBPL=USER      CALLER=0002A58C
16:41:25 RELEASED BYTES=000007B8 ADDR=003E8070 SUBPL=USER      CALLER=0002A640
16:41:25 RELEASED BYTES=00000878 ADDR=00304788 SUBPL=USER      CALLER=0002A698
16:41:25 RELEASED BYTES=000000B0 ADDR=002D0000 SUBPL=USER      CALLER=0002A768
16:41:25 RELEASED BYTES=000000C8 ADDR=002F37E8 SUBPL=USER      CALLER=00029EE0
16:41:26 OBTAINED BYTES=000000C8 ADDR=003E8760 SUBPL=USER      CALLER=000240BC
16:41:26 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00028534
16:41:26 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=000286D0
16:41:26 OBTAINED BYTES=000000B0 ADDR=003E86B0 SUBPL=USER      CALLER=000241B0
16:41:26 OBTAINED BYTES=000000A8 ADDR=003E8608 SUBPL=USER      CALLER=0002B25C
16:41:26 OBTAINED BYTES=00000870 ADDR=002F3040 SUBPL=USER      CALLER=0002B3F0
16:41:26 OBTAINED BYTES=000007B0 ADDR=002C1850 SUBPL=USER      CALLER=0002B610
16:41:26 RELEASED BYTES=000000A8 ADDR=003E8608 SUBPL=USER      CALLER=00029EE0
16:41:26 OBTAINED BYTES=000007B0 ADDR=003CD748 SUBPL=USER      CALLER=00024318
16:41:26 OBTAINED BYTES=000002C8 ADDR=003E83E8 SUBPL=USER      CALLER=000243A4
16:41:26 RELEASED BYTES=000000C8 ADDR=003E8760 SUBPL=USER      CALLER=00029EE0
16:41:26 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=000C0104
16:41:26 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=000C03E0
16:41:30 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BCD0
16:41:30 OBTAINED BYTES=00000050 ADDR=003E8398 SUBPL=USER      CALLER=0002DBE8
16:41:30 OBTAINED BYTES=00000128 ADDR=003E8270 SUBPL=USER      CALLER=00031ED4
16:41:30 RELEASED BYTES=00000050 ADDR=003E8398 SUBPL=USER      CALLER=0002E0F8
```

```
16:41:30 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:30 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:30 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:30 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:30 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:30 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:30 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:30 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=0002C028
16:41:31 RELEASED BYTES=000000A0 ADDR=003E8348 SUBPL=USER      CALLER=00029EE0
16:41:31 OBTAINED BYTES=00000098 ADDR=003E8350 SUBPL=USER      CALLER=0002CA78
16:41:31 OBTAINED BYTES=00000058 ADDR=003B5628 SUBPL=USER      CALLER=0002F114
16:41:31 RELEASED BYTES=00000058 ADDR=003B5628 SUBPL=USER      CALLER=0002F324
16:41:31 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002D6F4
16:41:31 OBTAINED BYTES=00000058 ADDR=003E82F8 SUBPL=USER      CALLER=00031864
16:41:31 OBTAINED BYTES=000007B0 ADDR=00396850 SUBPL=USER      CALLER=000318DC
16:41:31 RELEASED BYTES=00000058 ADDR=003E82F8 SUBPL=USER      CALLER=00031B48
16:41:31 OBTAINED BYTES=00000128 ADDR=003E8228 SUBPL=USER      CALLER=00031ED4
16:41:31 OBTAINED BYTES=000000E0 ADDR=003E8270 SUBPL=USER      CALLER=000327C4
16:41:34 RELEASED BYTES=00000028 ADDR=003B5240 SUBPL=USER      CALLER=00032CB8
16:41:34 OBTAINED BYTES=00000128 ADDR=003E8228 SUBPL=USER      CALLER=00031ED4
16:41:34 RELEASED BYTES=000007B0 ADDR=00396850 SUBPL=USER      CALLER=0002DFBC
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002E0F8
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002F5FC
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002F6D4
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002F9A4
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002FE74
16:41:34 RELEASED BYTES=00000098 ADDR=003E8350 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
```

```
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0C02BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002BE7C
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002D31C
16:41:34 OBTAINED BYTES=00000050 ADDR=003E8398 SUBPL=USER      CALLER=0002F9A4
16:41:34 RELEASED BYTES=00000050 ADDR=003E8398 SUBPL=USER      CALLER=0002FE74
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=000000C8 ADDR=003E8320 SUBPL=USER      CALLER=000246A0
16:41:34 RELEASED BYTES=000002C8 ADDR=003E83E8 SUBPL=USER      CALLER=0002A534
16:41:34 RELEASED BYTES=000007B0 ADDR=003CD748 SUBPL=USER      CALLER=0002A58C
16:41:34 RELEASED BYTES=000007B0 ADDR=002C1850 SUBPL=USER      CALLER=0002A640
16:41:34 RELEASED BYTES=00000870 ADDR=002F3040 SUBPL=USER      CALLER=0002A698
16:41:34 RELEASED BYTES=000000B0 ADDR=003E86B0 SUBPL=USER      CALLER=0002A768
16:41:34 RELEASED BYTES=000000C8 ADDR=003E8320 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002BCD0
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002DBE8
16:41:34 OBTAINED BYTES=00000128 ADDR=003E8700 SUBPL=USER      CALLER=00031ED4
16:41:34 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002E0F8
16:41:34 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=0002C028
16:41:34 RELEASED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=0002C028
16:41:34 RELEASED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=0002C028
16:41:34 RELEASED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=0002C028
16:41:34 RELEASED BYTES=000000A0 ADDR=003E8788 SUBPL=USER      CALLER=00029EE0
16:41:34 OBTAINED BYTES=00000098 ADDR=003E8790 SUBPL=USER      CALLER=0002CA78
16:41:34 OBTAINED BYTES=00000058 ADDR=003B5128 SUBPL=USER      CALLER=0002F114
16:41:34 RELEASED BYTES=00000058 ADDR=003B5128 SUBPL=USER      CALLER=0002F324
16:41:34 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002D8F4
16:41:34 OBTAINED BYTES=00000058 ADDR=003B5628 SUBPL=USER      CALLER=00031864
16:41:34 OBTAINED BYTES=00000790 ADDR=003E8000 SUBPL=USER      CALLER=000318DC
16:41:34 RELEASED BYTES=00000058 ADDR=003B5628 SUBPL=USER      CALLER=00031B48
16:41:34 OBTAINED BYTES=00000128 ADDR=002C1ED8 SUBPL=USER      CALLER=00031ED4
16:41:34 OBTAINED BYTES=000000E0 ADDR=002C1F20 SUBPL=USER      CALLER=000327C4
16:41:35 RELEASED BYTES=00000028 ADDR=003B5108 SUBPL=USER      CALLER=00032CB8
16:41:35 OBTAINED BYTES=00000128 ADDR=002C1ED8 SUBPL=USER      CALLER=00031ED4
16:41:35 RELEASED BYTES=00000790 ADDR=003E8000 SUBPL=USER      CALLER=0002DFBC
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002E0F8
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002F5FC
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002F6D4
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002F9A4
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002FE74
16:41:35 RELEASED BYTES=00000098 ADDR=003E8790 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002BE7C
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002BE7C
```

```
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002BE7C
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002BE7C
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=0002D31C
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002F9A4
16:41:35 RELEASED BYTES=00000050 ADDR=003B5630 SUBPL=USER      CALLER=0002FE74
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 OBTAINED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00024828
16:41:35 RELEASED BYTES=000002C0 ADDR=002D05C0 SUBPL=USER      CALLER=0002A534
16:41:35 RELEASED BYTES=00000798 ADDR=003E8828 SUBPL=USER      CALLER=0002A58C
16:41:35 RELEASED BYTES=00000790 ADDR=003A3020 SUBPL=USER      CALLER=0002A640
16:41:35 RELEASED BYTES=00000850 ADDR=003A37B0 SUBPL=USER      CALLER=0002A698
16:41:35 RELEASED BYTES=000000B0 ADDR=003B5180 SUBPL=USER      CALLER=0002A768
16:41:35 RELEASED BYTES=00002010 ADDR=003A1000 SUBPL=USER      CALLER=00024984
16:41:35 RELEASED BYTES=00000050 ADDR=003B5130 SUBPL=USER      CALLER=00029EE0
16:41:35 RELEASED BYTES=00002000 ADDR=003C9000 SUBPL=DMSOS000 CALLER=0039A884
16:41:36 RELEASED BYTES=00000338 ADDR=003CECC8 SUBPL=DMSOS000 CALLER=003E679C
16:41:36 RELEASED BYTES=00000400 ADDR=003CE000 SUBPL=DMSOS000 CALLER=000283D2
16:41:36 RELEASED BYTES=00000108 ADDR=003CDEF8 SUBPL=USER      CALLER=0000E286
```

END OF FIGURE 13.

Dynamic Memory Usuage.

BEGINNING OF FIGURE 14.

```
CICS     SD 00020000          RMODE 24  AMODE ANY
Invalid card - XM200030 203 SPE XMEDIT PROFILE, additional options.
Invalid card - *         MSUBS01  XM200030 B1 XM2292  11/06/89   17:23:27
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS01  ASSEMBLE A1 XM2292   6/28/89   14:46:19
MSUBS01  SD 00023EB0          RMODE ANY AMODE ANY
MLOADX       00023F1C
MLOAD        0002400E
MPURGE       000245F4
MEXIT        0002477C
CICSSCRN SD 00024C08          RMODE 24  AMODE ANY
IGZEBST  SD 00028088          RMODE ANY AMODE 31
IGZEBS2      00028316
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBINIT ASSEMBLE A1 XM2292   6/28/89   14:45:52
MSUBINIT SD 000284B0          RMODE ANY AMODE ANY
DEVINIT      00028708
MSUBEND      000288E0
Invalid card - XM200006 201 Incorrect terminal buffer column size on wide terminals
Invalid card - *         MSUBS44  XM200006 B1 XM2292  07/20/89   14:23:29
Invalid card - XM200029 203 SPE MTBFLG - Text borders by default flag.
Invalid card - *         MSUBS44  XM200029 B1 XM2292  09/25/89   16:15:20
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBS44  XM200065 B1 XM2292  01/12/90   14:20:58
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS44  ASSEMBLE A1 XM2292   7/12/89   16:45:39
MSUBS44  SD 00028978          RMODE ANY AMODE ANY
MDEFVS       000289A0
MPURVS       00028CB2
MDEFW        00028DE2
MPURW        0002916C
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS30  ASSEMBLE A1 XM2292   6/28/89   14:47:33
MSUBS30  SD 000295F0          RMODE ANY AMODE ANY
MSETAI       00029618
MCLRAI       00029740
MQRYAI       00029850
Invalid card - XM200001 201 Maintenance number change
```

```
Invalid card - *         MSUBCOM  XM200001 B1 XM2292  07/18/89   12:04:04
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCOM  ASSEMBLE A1 XM2292   7/08/89   18:55:36
MSUBCOM  SD 00029D58          RMODE ANY AMODE ANY
XMENUVER     00029E5A
MENUSUBS     00029D58
MVERS        00029D76
MSUBEXIT     00029E68
MSUBL2UP     0002AB22
.000001  SD 0002AC68          RMODE 24  AMODE 24
MSUBDATA     0002AC68
MSUBDBGE     0002AD30
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBXMRD ASSEMBLE A1 XM2292   6/28/89   14:48:19
MSUBXMRD SD 0002B1B8          RMODE ANY AMODE ANY
MREAD        0002B1DC
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS18  ASSEMBLE A1 XM2292   6/28/89   14:46:59
MSUBS18  SD 0002B868          RMODE ANY AMODE ANY
MSCRSZ       0002B8D4
MCTYPE       0002BA60
MCLSCR       0002BC22
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS09  ASSEMBLE A1 XM2292   6/28/89   14:46:42
MSUBS09  SD 0002BDA8          RMODE ANY AMODE ANY
MSCR2D       0002BDD0
MD2SCR       0002BF7C
MLD2S        0002C130
MLS2D        0002C28E
MTSTF        0002C458
MCLRF        0002C590
Invalid card - XM200052 203 Fix NOUNLOCK.
Invalid card - *         MSUBS02  XM200052 B1 XM2292  11/03/89   11:53:47
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBS02  XM200065 B1 XM2292  01/10/90   13:27:50
Invalid card - XM200066 204 01/10/90 PROG 4 in MSUBCMPC label NOSPEC1.
Invalid card - *         MSUBS02  XM200066 B1 XM2292  01/10/90   13:29:00
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
```

```
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS02  ASSEMBLE A1 XM2292   7/11/89   18:35:03
MSUBS02   SD 0002C6F8           RMODE ANY AMODE ANY
MDSPLY       0002C720
MDSPRD       0002C9CC
MRDSPC       0002CE9A
MRDSPR       0002CF8E
Invalid card - XM200037 203 MRSHOW ignored by windowing.
Invalid card - *          MSUBS02A XM200037 B1 XM2292  10/13/89   15:00:55
Invalid card - *          XMENUSRC MACLIB   A1 XM2292  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS02A ASSEMBLE A1 XM2292   6/28/89   14:46:24
MSUBS02A SD 0002D248            RMODE ANY AMODE ANY
MKEYP        0002D270
MCURP        0002D3AC
MPCUR        0002D50C
MRSHOW       0002D6A4
Invalid card - XM200024 203 SPE Border character/attribute definitions for MENUEXEC.
Invalid card - *          MSUBFSCR XM200024 B1 XM2292  09/26/89   14:32:41
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBFSCR XM200065 B1 XM2292  01/17/90   07:23:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBFSCR ASSEMBLE A1 XM2292   6/28/89   14:45:49
MSUBFSCR SD 0002D7D8            RMODE 24  AMODE 24
FSCWRITE     0002D7D8
FSCWREAD     0002D87E
FSCWSFR      0002D9B6
FSCWSF       0002DA8E
FSCERASE     0002DB72
FSCRBUFF     0002DC22
FSCRMOD      0002DCE2
FSCREAD      0002DDB8
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *          MSUBPQSF XM200025 B1 XM2292  09/12/89   10:32:27
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBPQSF XM200065 B1 XM2292  01/17/90   08:04:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBPQSF ASSEMBLE A1 XM2292   6/28/89   14:46:03
MSUBPQSF SD 0002E228            RMODE ANY AMODE ANY
CVTPQSF      0002E228
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBTMCH XM200065 B1 XM2292  01/17/90   07:50:42
Invalid card - XM200068 204 01/17/90 Diag 8C Error codes on Rx+1, not Ry+1.
Invalid card - *          MSUBTMCH XM200068 B1 XM2292  01/17/90   07:55:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
```

```
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTMCH ASSEMBLE A1 XM2292   6/28/89   14:48:14
MSUBTMCH SD 0002E6B8            RMODE ANY AMODE ANY
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTBL5 ASSEMBLE A1 XM2292   6/28/89   14:48:12
MSUBTBL5 SD 0002E988            RMODE 24  AMODE 24
XTBLRG       0002E988
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTBL4 ASSEMBLE A1 XM2292   6/28/89   14:48:11
MSUBTBL4 SD 0002EA88            RMODE 24  AMODE 24
XTBLGT       0002EA88
XTBLGR       0002EB88
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBSCRB ASSEMBLE A1 XM2292   6/28/89   14:46:11
MSUBSCRB SD 0002EC88            RMODE 24  AMODE 24
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBINSA XM200065 B1 XM2292  01/17/90   10:20:11
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBINSA ASSEMBLE A1 XM2292   6/28/89   14:45:57
MSUBINSA SD 0002F090            RMODE ANY AMODE ANY
INSERTSA     0002F090
TRASHSA      0002F576
Invalid card - XM200017 202 Preserve last user input in background windows.
Invalid card - *          MSUBINPS XM200017 B1 XM2292  08/13/89   17:09:53
Invalid card - XM200045 203 SPE MARRW, MARRWR window array subroutines.
Invalid card - *          MSUBINPS XM200045 B1 XM2292  10/26/89   16:38:57
Invalid card - *          XMENUSRC MACLIB   A1 XM2292  10/19/89   15:43:36
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBINPS ASSEMBLE A1 XM2292   6/28/89   14:45:55
MSUBINPS SD 0002F738            RMODE ANY AMODE ANY
MSUBOUTF     0002F738
MSUBKEYP     0002F920
MSUBSPEN     0002F9FC
MSUBINPF     0002FC98
PFKTAB       0002FE9C
```

```
PFKNAMES    0002FEDC
Invalid card - XM200004 201 Occasional screen errors, screen not redrawn after MORE...
Invalid card - *         MSUBWINF XM200004 B1 XM2292  07/19/89   16:52:00
Invalid card - XM200017 202 Preserve last user input in background windows.
Invalid card - *         MSUBWINF XM200017 B1 XM2292  08/13/89   17:03:20
Invalid card - XM200050 203 Support SA orders in XMENU windows.
Invalid card - *         MSUBWINF XM200050 B1 XM2292  11/02/89   15:23:22
Invalid card - XM200051 203 Support APL/TEXT in background XMENU windows.
Invalid card - *         MSUBWINF XM200051 B1 XM2292  11/02/89   16:44:49
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBWINF XM200065 B1 XM2292  01/10/90   11:18:15
Invalid card - XM200067 204 01/12/90 Free storage abend. Obtain length not set.
Invalid card - *         MSUBWINF XM200067 B1 XM2292  01/12/90   13:52:55
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBWINF ASSEMBLE A1 XM2292   7/12/89   17:22:41
MSUBWINF SD 000300A8          RMODE 24  AMODE 24
WINWRITE    000300A8
WINWREAD    00030918
WINRMOD     00030970
.000002  SD 00030C60          RMODE 24  AMODE 24
ANFEJPMB    00030CA8
Invalid card - SLC  000000    ANFEJPMB
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *         MSUBRMEX XM200025 B1 XM2292  09/12/89   14:26:53
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBRMEX XM200065 B1 XM2292  01/15/90   13:17:37
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBRMEX ASSEMBLE A1 XM2292   6/28/89   14:46:06
MSUBRMEX SD 00030D68          RMODE ANY AMODE ANY
RM3279      00030D68
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBCMPC XM200065 B1 XM2292  01/17/90   10:19:16
Invalid card - XM200069 204 01/18/90 Error using EDS menu on 3277 type term.
Invalid card - *         MSUBCMPC XM200069 B1 XM2292  01/19/90   09:39:40
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCMPC ASSEMBLE A1 XM2292   6/28/89   14:45:41
MSUBCMPC SD 00031170          RMODE 24  AMODE 24
CVTOUT      00031170
CVTIN       0003146C
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBOPT  ASSEMBLE A1 XM2292   6/28/89   14:46:01
MSUBOPT  SD 000317E0          RMODE ANY AMODE ANY
```

```
OPTIMIZE    000317E0
Invalid card - XM200034 203 Prot. excp. stacking lines on MORE under old CMS.
Invalid card - *         MSUBSCIO XM200034 B1 XM2292  10/10/89   10:14:48
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBSCIO ASSEMBLE A1 XM2292   6/28/89   14:46:10
MSUBSCIO SD 00031E50          RMODE ANY AMODE ANY
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBWAIT ASSEMBLE A1 XM2292   6/28/89   14:48:17
MSUBWAIT SD 00032740          RMODE ANY AMODE ANY
Invalid card - XM200050 203 Support SA orders in XMENU windows.
Invalid card - *         MSUBCVTA XM200050 B1 XM2292  11/02/89   10:34:58
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *         MSUBCVTA XM200065 B1 XM2292  01/18/90   09:01:11
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTA ASSEMBLE A1 XM2292   7/12/89   16:45:36
MSUBCVTA SD 00032D50          RMODE 24  AMODE 24
.000003  SD 000341E8          RMODE 24  AMODE 24
ATAEJPFB    00034230
Invalid card - SLC  000000    ATAEJPFB
Invalid card - XM200002 201 Fix title name length check
Invalid card - *         MSUBBLDS XM200002 B1 XM2292  07/17/89   16:21:14
Invalid card - XM200003 201 Unformatted screen problems
Invalid card - *         MSUBBLDS XM200003 B1 XM2292  07/18/89   12:00:42
Invalid card - XM200023 202 Remove unnecessary local flags cauing border problems.
Invalid card - *         MSUBBLDS XM200023 B1 XM2292  08/21/89   17:33:44
Invalid card - XM200039 203 No corners on borders without walls.
Invalid card - *         MSUBBLDS XM200039 B1 XM2292  10/13/89   16:20:54
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   S2 MNT19E  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         MSUBBLDS ASSEMBLE A1 XM2292   7/08/89   18:23:53
MSUBBLDS SD 00034280          RMODE 24  AMODE 24
.000004  SD 00034CE8          RMODE 24  AMODE 24
JDSNIQVA    00034D30
Invalid card - SLC  000000    JDSNIQVA
Invalid card - XM200003 201 Unformatted screen problems
Invalid card - *         MSUBBLDT XM200003 B1 XM2292  07/18/89   11:58:13
Invalid card - XM200022 202 Performance improvements in changed data calculations.
Invalid card - *         MSUBBLDT XM200022 B1 XM2292  08/21/89   14:03:25
Invalid card - XM200042 203 Totally unformatted VSCREEN > terminal size, no wrap attrib.
Invalid card - *         MSUBBLDT XM200042 B1 XM2292  10/16/89   15:18:35
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  10/14/89   15:35:26
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
```

```
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBBLDT ASSEMBLE A1 XM2292   6/28/89   14:45:35
MSUBBLDT SD 00034D78          RMODE 24  AMODE 24
.000005  SD 00035500          RMODE 24  AMODE 24
JDTQIPSA    00035548
Invalid card - SLC  000000    JDTQIPSA
Invalid card - XM200018 202 SPE Further data output optimization.
Invalid card - *         MSUBCVTB XM200018 B1 XM2292  08/13/89   17:36:08
Invalid card - XM200022 202 Performance improvements in changed data calculations.
Invalid card - *         MSUBCVTB XM200022 B1 XM2292  08/21/89   16:51:06
Invalid card - XM200024 203 SPE Border character/attribute definitions for MENUEXEC.
Invalid card - *         MSUBCVTB XM200024 B1 XM2292  10/26/89   17:31:54
Invalid card - XM200046 203 SPE MWCCF subroutine.
Invalid card - *         MSUBCVTB XM200046 B1 XM2292  10/26/89   17:32:39
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/26/89   17:25:28
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTB ASSEMBLE A1 XM2292   6/30/89   15:21:16
MSUBCVTB SD 00035598          RMODE 24  AMODE 24
.000006  SD 000363C8          RMODE 24  AMODE 24
JTBAIRGC    00036410
Invalid card - SLC  000000    JTBAIRGC
Invalid card - XM200003 201 Unformatted screen problems
Invalid card - *         MSUBCVTI XM200003 B1 XM2292  07/17/89   17:52:28
Invalid card - XM200056 203 Storage overlay when unformatted window < terminal size.
Invalid card - *         MSUBCVTI XM200056 B1 XM2292  11/27/89   12:11:36
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 XM2191  11/16/89   16:13:39
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTI ASSEMBLE A1 XM2292   6/28/89   14:45:46
MSUBCVTI SD 000364C8          RMODE 24  AMODE 24
.000007  SD 000379C8          RMODE 24  AMODE 24
KTIBIMLB    00037A10
Invalid card - SLC  000000    KTIBIMLB
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTJ ASSEMBLE A1 XM2292   6/28/89   14:45:47
MSUBCVTJ SD 00037A80          RMODE 24  AMODE 24
.000008  SD 00038490          RMODE 24  AMODE 24
JTJLITWA    000384D8
Invalid card - SLC  000000    JTJLITWA
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *         MSUBTBL3 XM200025 B1 XM2292  09/12/89   10:34:46
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL3 ASSEMBLE A1 XM2292   6/28/89   14:48:11
```

```
MSUBTBL3 SD 00038520          RMODE 24  AMODE 24
RMTRTTAB    00038520
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL2 ASSEMBLE A1 XM2292   6/28/89   14:48:10
MSUBTBL2 SD 00038620          RMODE 24  AMODE 24
TBLRG       00038620
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL1 ASSEMBLE A1 XM2292   6/28/89   14:48:10
MSUBTBL1 SD 00038720          RMODE 24  AMODE 24
TBLGT       00038720
TBLGR       00038820
TBLZO       00038920
TBLZI       00038A20
TBLTO       00038B20
TBLTI       00038D20
TBLXO       00039120
TBLXI       00039220
TBLAO       00038F20
TBLAI       00039020
```

END OF FIGURE 14. DYNAMIC LOAD MAP

BEGINNING OF FIGURE 15.

```
CICS     SD 00020000          RMODE 24  AMODE ANY
Invalid card - XM200030 203 SPE XMEDIT PROFILE, additional options.
Invalid card - *         MSUBS01  XM200030 B1 XM2292  11/06/89   17:23:27
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS01  ASSEMBLE A1 XM2292   6/28/89   14:46:19
MSUBS01  SD 00023EF0          RMODE ANY AMODE ANY
MLOADX       00023F5C
MLOAD        0002404E
MPURGE       00024634
MEXIT        000247BC
CICSSCRN SD 00024C48          RMODE 24  AMODE ANY
CPPO0010 SD 000280C8          RMODE 24  AMODE ANY
CPPO0011 SD 0005BD78          RMODE 24  AMODE ANY
CPPO0015 SD 0006CEA8          RMODE 24  AMODE ANY
CPPO0014 SD 00083198          RMODE 24  AMODE ANY
CPPO0029 SD 0009CED8          RMODE 24  AMODE ANY
IGZEBST  SD 000C6F30          RMODE ANY AMODE 31
IGZEBS2      000C71BE
Invalid card - *         XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *         XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBINIT ASSEMBLE A1 XM2292   6/28/89   14:45:52
MSUBINIT SD 000C7358          RMODE ANY AMODE ANY
DEVINIT      000C75B0
MSUBEND      000C7788
Invalid card - XM200006 201 Incorrect terminal buffer column size on wide terminals
Invalid card - *          MSUBS44  XM200006 B1 XM2292  07/20/89   14:23:29
Invalid card - XM200029 203 SPE MTBFLG - Text borders by default flag.
Invalid card - *          MSUBS44  XM200029 B1 XM2292  09/25/89   16:15:20
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBS44  XM200065 B1 XM2292  01/12/90   14:20:58
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS44  ASSEMBLE A1 XM2292   7/12/89   16:45:39
MSUBS44  SD 000C7820          RMODE ANY AMODE ANY
MDEFVS       000C7848
MPURVS       000C7B5A
MDEFW        000C7C8A
MPURW        000C8014
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS30  ASSEMBLE A1 XM2292   6/28/89   14:47:33
```

```
MSUBS30  SD 000C8498          RMODE ANY AMODE ANY
MSETAI       000C84C0
MCLRAI       000C85E8
MQRYAI       000C86F8
Invalid card - XM200001 201 Maintenance number change
Invalid card - *          MSUBCOM  XM200001 B1 XM2292  07/18/89   12:04:04
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBCOM  ASSEMBLE A1 XM2292   7/08/89   18:55:36
MSUBCOM  SD 000C8C00          RMODE ANY AMODE ANY
XMENUVER     000C8D02
MENUSUBS     000C8C00
MVERS        000C8C1E
MSUBEXIT     000C8D10
MSUBL2UP     000C99CA
.000017  SD 000C9B10          RMODE 24  AMODE 24
MSUBDATA     000C9B10
MSUBDBGE     000C9BD8
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBXMRD ASSEMBLE A1 XM2292   6/28/89   14:48:19
MSUBXMRD SD 000CA060          RMODE ANY AMODE ANY
MREAD        000CA084
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS18  ASSEMBLE A1 XM2292   6/28/89   14:46:59
MSUBS18  SD 000CA710          RMODE ANY AMODE ANY
MSCRSZ       000CA77C
MCTYPE       000CA908
MCLSCR       000CAACA
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBS09  ASSEMBLE A1 XM2292   6/28/89   14:46:42
MSUBS09  SD 000CAC50          RMODE ANY AMODE ANY
MSCR2D       000CAC78
MD2SCR       000CAE24
MLD2S        000CAFD8
MLS2D        000CB136
MTSTF        000CB300
MCLRF        000CB438
Invalid card - XM200052 203 Fix NOUNLOCK.
Invalid card - *          MSUBS02  XM200052 B1 XM2292  11/03/89   11:53:47
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBS02  XM200065 B1 XM2292  01/10/90   13:27:50
Invalid card - XM200066 204 01/10/90 PROG 4 in MSUBCMPC label NOSPEC1.
Invalid card - *          MSUBS02  XM200066 B1 XM2292  01/10/90   13:29:00
```

```
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS02  ASSEMBLE A1 XM2292   7/11/89   18:35:03
MSUBS02  SD 000CB5A0           RMODE ANY AMODE ANY
MDSPLY      000CB5C8
MDSPRD      000CB874
MRDSPC      000CBD42
MRDSPR      000CBE36
Invalid card - XM200037 203 MRSHOW ignored by windowing.
Invalid card - *          MSUBS02A XM200037 B1 XM2292  10/13/89   15:00:55
Invalid card - *          XMENUSRC MACLIB   A1 XM2292  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS02A ASSEMBLE A1 XM2292   6/28/89   14:46:24
MSUBS02A SD 000CC0F0           RMODE ANY AMODE ANY
MKEYP       000CC118
MCURP       000CC254
MPCUR       000CC3B4
MRSHOW      000CC54C
ARIPADR4 SD 000CC680           RMODE 24  AMODE 24
ARIPADR5    000CC78E
SCRN1120 SD 000CC7B0           RMODE 24  AMODE ANY
ARIRVST  SD 000D1E20           RMODE ANY AMODE 31
ARIPRDI     000D1E20
SCRN1121 SD 000D28F0           RMODE 24  AMODE ANY
SCRN1141 SD 000D8DA0           RMODE 24  AMODE ANY
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBS12  ASSEMBLE A1 XM2292   6/28/89   14:46:48
MSUBS12  SD 000E09C0           RMODE ANY AMODE ANY
MQATTR      000E09E4
MFQRY       000E0AAC
MFSET       000E0CAC
SCRN1140 SD 000E0FB0           RMODE 24  AMODE ANY
SCRN2460 SD 000EBF00           RMODE 24  AMODE ANY
Invalid card - XM200024 203 SPE Border character/attribute definitions for MENUEXEC.
Invalid card - *          MSUBFSCR XM200024 B1 XM2292  09/26/89   14:32:41
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBFSCR XM200065 B1 XM2292  01/17/90   07:23:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBFSCR ASSEMBLE A1 XM2292   6/28/89   14:45:49
MSUBFSCR SD 000EFB70           RMODE 24  AMODE 24
FSCWRITE    000EFB70
FSCWREAD    000EFC16
FSCWSFR     000EFD4E
FSCWSF      000EFE26
```

```
FSCERASE    000EFF0A
FSCRBUFF    000EFFBA
FSCRMOD     000F007A
FSCREAD     000F0150
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *          MSUBPQSF XM200025 B1 XM2292  09/12/89   10:32:27
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBPQSF XM200065 B1 XM2292  01/17/90   08:04:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBPQSF ASSEMBLE A1 XM2292   6/28/89   14:46:03
MSUBPQSF SD 000F05C0           RMODE ANY AMODE ANY
CVTPQSF     000F05C0
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBTMCH XM200065 B1 XM2292  01/17/90   07:50:42
Invalid card - XM200068 204 01/17/90 Diag 8C Error codes on Rx+1, not Ry+1.
Invalid card - *          MSUBTMCH XM200068 B1 XM2292  01/17/90   07:55:10
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTMCH ASSEMBLE A1 XM2292   6/28/89   14:48:14
MSUBTMCH SD 000F0A50           RMODE ANY AMODE ANY
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTBL5 ASSEMBLE A1 XM2292   6/28/89   14:48:12
MSUBTBL5 SD 000F0D20           RMODE 24  AMODE 24
XTBLRG      000F0D20
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBTBL4 ASSEMBLE A1 XM2292   6/28/89   14:48:11
MSUBTBL4 SD 000F0E20           RMODE 24  AMODE 24
XTBLGT      000F0E20
XTBLGR      000F0F20
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBSCRB ASSEMBLE A1 XM2292   6/28/89   14:46:11
MSUBSCRB SD 000F1020           RMODE 24  AMODE 24
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBINSA XM200065 B1 XM2292  01/17/90   10:20:11
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
```

```
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBINSA ASSEMBLE A1 XM2292   6/28/89   14:45:57
MSUBINSA SD 000F1428           RMODE ANY AMODE ANY
INSERTSA    000F1428
TRASHSA     000F190E
Invalid card - XM200017 202 Preserve last user input in background windows.
Invalid card - *          MSUBINPS XM200017 B1 XM2292  08/13/89   17:09:53
Invalid card - XM200045 203 SPE MARRW, MARRWR window array subroutines.
Invalid card - *          MSUBINPS XM200045 B1 XM2292  10/26/89   16:38:57
Invalid card - *          XMENUSRC MACLIB   A1 XM2292  10/19/89   15:43:36
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBINPS ASSEMBLE A1 XM2292   6/28/89   14:45:55
MSUBINPS SD 000F1AD0           RMODE ANY AMODE ANY
MSUBOUTF    000F1AD0
MSUBKEYF    000F1CB8
MSUBSPEN    000F1D94
MSUBINPF    000F2030
PFKTAB      000F2234
PFKNAMES    000F2274
Invalid card - XM200004 201 Occasional screen errors, screen not redrawn after MORE...
Invalid card - *          MSUBWINF XM200004 B1 XM2292  07/19/89   16:52:00
Invalid card - XM200017 202 Preserve last user input in background windows.
Invalid card - *          MSUBWINF XM200017 B1 XM2292  08/13/89   17:03:20
Invalid card - XM200050 203 Support SA orders in XMENU windows.
Invalid card - *          MSUBWINF XM200050 B1 XM2292  11/02/89   15:23:22
Invalid card - XM200051 203 Support APL/TEXT in background XMENU windows.
Invalid card - *          MSUBWINF XM200051 B1 XM2292  11/02/89   16:44:49
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBWINF XM200065 B1 XM2292  01/10/90   11:18:15
Invalid card - XM200067 204 01/12/90 Free storage abend. Obtain length not set.
Invalid card - *          MSUBWINF XM200067 B1 XM2292  01/12/90   13:52:55
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBWINF ASSEMBLE A1 XM2292   7/12/89   17:22:41
MSUBWINF SD 000F2440           RMODE 24  AMODE 24
WINWRITE    000F2440
WINWREAD    000F2CB0
WINRMOD     000F2D08
.000018  SD 000F2FF8           RMODE 24  AMODE 24
ANFEJPMB    000F3040
Invalid card - SLC  000000    ANFEJPMB
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *          MSUBRMEX XM200025 B1 XM2292  09/12/89   14:26:53
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBRMEX XM200065 B1 XM2292  01/15/90   13:17:37
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBRMEX ASSEMBLE A1 XM2292   6/28/89   14:46:06
```

```
MSUBRMEX SD 000F3100           RMODE ANY AMODE ANY
RM3279      000F3100
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBCMPC XM200065 B1 XM2292  01/17/90   10:19:16
Invalid card - XM200069 204 01/18/90 Error using EDS menu on 3277 type term.
Invalid card - *          MSUBCMPC XM200069 B1 XM2292  01/19/90   09:39:40
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBCMPC ASSEMBLE A1 XM2292   6/28/89   14:45:41
MSUBCMPC SD 000F3508           RMODE 24  AMODE 24
CVTOUT      000F3508
CVTIN       000F3804
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBOPT  ASSEMBLE A1 XM2292   6/28/89   14:46:01
MSUBOPT  SD 000F3B78           RMODE ANY AMODE ANY
OPTIMIZE    000F3B78
Invalid card - XM200034 203 Prot. excp. stacking lines on MORE under old CMS.
Invalid card - *          MSUBSCIO XM200034 B1 XM2292  10/10/89   10:14:48
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBSCIO ASSEMBLE A1 XM2292   6/28/89   14:46:10
MSUBSCIO SD 000F41E8           RMODE ANY AMODE ANY
Invalid card - *          XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *          XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *          KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *          DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBWAIT ASSEMBLE A1 XM2292   6/28/89   14:48:17
MSUBWAIT SD 000F4AD8           RMODE ANY AMODE ANY
Invalid card - XM200050 203 Support SA orders in XMENU windows.
Invalid card - *          MSUBCVTA XM200050 B1 XM2292  11/02/89   10:34:58
Invalid card - XM200065 204 01/09/90 Documentation Changes.
Invalid card - *          MSUBCVTA XM200065 B1 XM2292  01/18/90   09:01:11
Invalid card - *          XMENUSRC MACLIB   A1 XM2191   1/09/90    8:36:48
Invalid card - *          XMENU    MACLIB   A2 XM2292   1/09/90    8:43:29
Invalid card - *          KCOMMAC  MACLIB   D1 KCM292   1/26/90    7:07:53
Invalid card - *          DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *          CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *          OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *          DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *          MSUBCVTA ASSEMBLE A1 XM2292   7/12/89   16:45:36
MSUBCVTA SD 000F50E8           RMODE 24  AMODE 24
.000019  SD 000F6580           RMODE 24  AMODE 24
ATAEJPFB    000F65C8
Invalid card - SLC  000000    ATAEJPFB
Invalid card - XM200002 201 Fix title name length check
Invalid card - *          MSUBBLDS XM200002 B1 XM2292  07/17/89   16:21:14
Invalid card - XM200003 201 Unformatted screen problems
```

```
Invalid card - *         MSUBBLDS XM200003 B1 XM2292  07/18/89   12:00:42
Invalid card - XM200023 202 Remove unnecessary local flags causing border problems.
Invalid card - *         MSUBBLDS XM200023 B1 XM2292  08/21/89   17:33:44
Invalid card - XM200039 203 No corners on borders without walls.
Invalid card - *         MSUBBLDS XM200039 B1 XM2292  10/13/89   16:20:54
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   S2 MNT19E  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         MSUBBLDS ASSEMBLE A1 XM2292   7/08/89   18:23:53
MSUBBLDS SD 000F6618          RMODE 24  AMODE 24
.000020  SD 000F7080          RMODE 24  AMODE 24
JDSNIQVA    000F70C8
Invalid card - SLC  000000    JDSNIQVA
Invalid card - XM200003 201 Unformatted screen problems
Invalid card - *         MSUBBLDT XM200003 B1 XM2292  07/18/89   11:58:13
Invalid card - XM200022 202 Performance improvements in changed data calculations.
Invalid card - *         MSUBBLDT XM200022 B1 XM2292  08/21/89   14:03:25
Invalid card - XM200042 203 Totally unformatted VSCREEN > terminal size, no wrap attrib.
Invalid card - *         MSUBBLDT XM200042 B1 XM2292  10/16/89   15:18:35
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  10/14/89   15:35:26
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBBLDT ASSEMBLE A1 XM2292   6/28/89   14:45:35
MSUBBLDT SD 000F7110          RMODE 24  AMODE 24
.000021  SD 000F7898          RMODE 24  AMODE 24
JDTQIPSA    000F78E0
Invalid card - SLC  000000    JDTQIPSA
Invalid card - XM200018 202 SPE Further data output optimization.
Invalid card - *         MSUBCVTB XM200018 B1 XM2292  08/13/89   17:36:08
Invalid card - XM200022 202 Performance improvements in changed data calculations.
Invalid card - *         MSUBCVTB XM200022 B1 XM2292  08/21/89   16:51:06
Invalid card - XM200024 203 SPE Border character/attribute definitions for MENUEXEC.
Invalid card - *         MSUBCVTB XM200024 B1 XM2292  10/26/89   17:31:54
Invalid card - XM200046 203 SPE MWCCF subroutine.
Invalid card - *         MSUBCVTB XM200046 B1 XM2292  10/26/89   17:32:39
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/26/89   17:25:28
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTB ASSEMBLE A1 XM2292   6/30/89   15:21:16
MSUBCVTB SD 000F7930          RMODE 24  AMODE 24
.000022  SD 000F8760          RMODE 24  AMODE 24
JTBAIRGC    000F87A8
Invalid card - SLC  000000    JTBAIRGC
Invalid card - XM200003 201 Unformatted screen problems
Invalid card - *         MSUBCVTI XM200003 B1 XM2292  07/17/89   17:52:28
Invalid card - XM200056 203 Storage overlay when unformatted window < terminal size.
Invalid card - *         MSUBCVTI XM200056 B1 XM2292  11/27/89   12:11:36
Invalid card - *         XMENUSRC MACLIB   A1 XM2292  11/03/89   14:16:41
Invalid card - *         XMENU    MACLIB   A2 XM2292  11/01/89   10:53:53
Invalid card - *         KCOMMAC  MACLIB   A1 XM2191  11/16/89   16:13:39
Invalid card - *         DMSSP    MACLIB   S2 XA2190   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2190   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2190  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTI ASSEMBLE A1 XM2292   6/28/89   14:45:46
```

```
MSUBCVTI SD 000F8860          RMODE 24  AMODE 24
.000023  SD 000F9D60          RMODE 24  AMODE 24
KTIBIMLB    000F9DA8
Invalid card - SLC  000000    KTIBIMLB
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBCVTJ ASSEMBLE A1 XM2292   6/28/89   14:45:47
MSUBCVTJ SD 000F9E18          RMODE 24  AMODE 24
.000024  SD 000FA828          RMODE 24  AMODE 24
JTJLITWA    000FA870
Invalid card - SLC  000000    JTJLITWA
Invalid card - XM200025 203 Remove GE characters if not supported by the terminal.
Invalid card - *         MSUBTBL3 XM200025 B1 XM2292  09/12/89   10:34:46
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL3 ASSEMBLE A1 XM2292   6/28/89   14:48:11
MSUBTBL3 SD 000FA8B8          RMODE 24  AMODE 24
RMTRTTAB    000FA8B8
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL2 ASSEMBLE A1 XM2292   6/28/89   14:48:10
MSUBTBL2 SD 000FA9B8          RMODE 24  AMODE 24
TBLRG       000FA9B8
Invalid card - *         XMENUSRC MACLIB   A1 XM2191  10/11/89   18:51:14
Invalid card - *         XMENU    MACLIB   A2 XM2292  10/05/89   18:33:27
Invalid card - *         KCOMMAC  MACLIB   A1 KCM292  10/02/89   16:39:44
Invalid card - *         DMSSP    MACLIB   S2 XA2490   6/25/89   12:57:01
Invalid card - *         CMSLIB   MACLIB   S2 XA2490   7/10/87   13:37:27
Invalid card - *         OSMACRO  MACLIB   S2 XA2490  10/16/87    0:45:33
Invalid card - *         DMKSP    MACLIB   A1 MN6194   8/19/88   14:35:47
Invalid card - *         MSUBTBL1 ASSEMBLE A1 XM2292   6/28/89   14:48:10
MSUBTBL1 SD 000FAAB8          RMODE 24  AMODE 24
TBLGT       000FAAB8
TBLGR       000FABB8
TBLZO       000FACB8
TBLZI       000FADB8
TBLTO       000FAEB8
TBLTI       000FB0B8
TBLXO       000FB4B8
TBLXI       000FB5B8
TBLAO       000FB2B8
TBLAI       000FB3B8
```

END OF FIGURE 15. STATIC LOAD MAP

BEGINNING OF FIGURE 16.

```
SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'BID '
TNAME CNAME        COLTYPE  LENGTH
----- ------------ -------- -------
BID   DATE_OF_BID  DATE
BID   EMP_NO       CHAR           5
BID   POINTS_BID   SMALLINT
BID   SCH_NO       CHAR           2
BID   SSN          CHAR           9
BID   STATUS       CHAR           8
BID   TIME_OF_BID  TIME
ELO2121I ******** End-of-Data **********



SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'INTERVIEW'
TNAME     CNAME           COLTYPE  LENGTH
--------- --------------- -------- -------
INTERVIEW ATTEND_PRENIGHT CHAR           1
INTERVIEW EDIT_DATE       DATE
INTERVIEW EDIT_TIME       TIME
INTERVIEW EMP_NO          CHAR           5
INTERVIEW INT_DATE        DATE
INTERVIEW INT_NO          CHAR           2
INTERVIEW INT_TIME        TIME
INTERVIEW SCH_NO          CHAR           2
INTERVIEW SSN             CHAR           9
INTERVIEW STATUS          CHAR           8
ELO2121I ******** End-of-Data **********



SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'INTERVIEW_SCHEDULE'
TNAME              CNAME           COLTYPE  LENGTH
------------------ --------------- -------- -------
INTERVIEW_SCHEDULE CLOSED_INT      CHAR           1
INTERVIEW_SCHEDULE CPPO_STAFF_ID   CHAR          11
INTERVIEW_SCHEDULE EDIT_DATE       DATE
INTERVIEW_SCHEDULE EDIT_TIME       TIME
INTERVIEW_SCHEDULE EMP_NO          CHAR           5
INTERVIEW_SCHEDULE INT_DATE        DATE
INTERVIEW_SCHEDULE INT_LENGTH      SMALLINT
INTERVIEW_SCHEDULE INT_NO          CHAR           2
INTERVIEW_SCHEDULE INTERVIEWER     CHAR          36
INTERVIEW_SCHEDULE NO_SLOTS        SMALLINT
INTERVIEW_SCHEDULE PASSWORD_CLOSED CHAR           8
INTERVIEW_SCHEDULE ROOM_NO         CHAR           2
INTERVIEW_SCHEDULE SCH_NO          CHAR           2
INTERVIEW_SCHEDULE SLOTS_TAKEN     SMALLINT
INTERVIEW_SCHEDULE START_TIME      TIME
INTERVIEW_SCHEDULE STATUS          CHAR           8
ELO2121I ******** End-of-Data **********



SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'VISIT'
TNAME CNAME              COLTYPE  LENGTH
```

```
----- ------------------ -------- -------
VISIT ADDITIONAL_REQMTS  CHAR          60
VISIT ADMIN_NOTES        CHAR         254
VISIT BID_BY_DATE        DATE
VISIT CITIZEN_NOTES      CHAR          40
VISIT CITIZEN_REQD       CHAR           1
VISIT CONFRMD_PRE_NIGHT  CHAR          10
VISIT CPPO_STAFF_ID      CHAR           9
VISIT DATE_PRE_NIGHT     CHAR          10
VISIT DATE_1_INTERVIEW   DATE
VISIT DATE_2_INTERVIEW   CHAR          10
VISIT DATE_3_INTERVIEW   CHAR          10
VISIT DAY1_NUM_ROOMS     INTEGER
VISIT DAY2_NUM_ROOMS     INTEGER
VISIT DAY3_NUM_ROOMS     INTEGER
VISIT DEGREE             CHAR           3
VISIT DEGREE_REQMT       CHAR           3
VISIT DIVISION           CHAR          40
VISIT EDIT_DATE          CHAR          10
VISIT EDIT_TIME          CHAR           8
VISIT EMP_NO             CHAR           5
VISIT EXCLUDED_MAJORS    CHAR          24
VISIT GPA_NOTES          CHAR          24
VISIT GRAD_WHEN          CHAR           3
VISIT INV_RCD_PRE_NIGHT  CHAR          10
VISIT LIT_1_RCVD_DATE    CHAR          10
VISIT LIT_1_TYPE         CHAR          28
VISIT LIT_2_RCVD_DATE    CHAR          10
VISIT LIT_2_TYPE         CHAR          28
VISIT LIT_3_RCVD_DATE    CHAR          10
VISIT LIT_3_TYPE         CHAR          28
VISIT LOCATION           CHAR          34
VISIT LOCN_PRE_NIGHT     CHAR          35
VISIT MAJORS             CHAR          48
VISIT MIN_MAJOR_GPA      DECIMAL  ( 5, 2)
VISIT MIN_OVERALL_GPA    DECIMAL  ( 5, 2)
VISIT NUM_CLOSED_SLOTS   INTEGER
VISIT NUM_OPEN_SLOTS     INTEGER
VISIT NUM_SLOTS_TAKEN    INTEGER
VISIT NUM_TOTAL_SLOTS    INTEGER
VISIT POINTS_USED_TO_WIN INTEGER
VISIT POSITION           CHAR          48
VISIT PRE_NGT_NOTES      CHAR          64
VISIT SCH_NO             CHAR           2
VISIT SCHEDULE_BY_DATE   DATE
VISIT STATUS             CHAR           8
VISIT TIME_PRE_NIGHT     CHAR           8
VISIT VIDEO_AVAIL        CHAR           1
VISIT VIDEO_RCVD_DATE    CHAR          10
VISIT WHO_CNFRMD_PRE_NGT CHAR          36
ELO2121I ******** End-of-Data **********



SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'EMPLOYER'
TNAME    CNAME         COLTYPE  LENGTH
-------- ------------- -------- -------
EMPLOYER ADDR1         CHAR          28
EMPLOYER ADDR2         CHAR          28
EMPLOYER CITY          CHAR          22
EMPLOYER CODE1         CHAR           3
EMPLOYER CODE2         CHAR           3
EMPLOYER CODE3         CHAR           3
EMPLOYER CODE4         CHAR           3
```

```
EMPLOYER CODE5          CHAR            3
EMPLOYER CODE6          CHAR            3
EMPLOYER CPPO_STAFF_ID  CHAR            9
EMPLOYER DIVISION       CHAR           30
EMPLOYER EDIT_DATE      DATE
EMPLOYER EDIT_TIME      TIME
EMPLOYER EMP_NO         CHAR            5
EMPLOYER FAX            CHAR           14
EMPLOYER FIRST_NAME     CHAR           13
EMPLOYER LAST_NAME      CHAR           22
EMPLOYER MEMO           CHAR          128
EMPLOYER MI             CHAR            1
EMPLOYER MR_MS          CHAR            4
EMPLOYER M1             CHAR            3
EMPLOYER M2             CHAR            3
EMPLOYER M3             CHAR            3
EMPLOYER M4             CHAR            3
EMPLOYER NAME           CHAR           40
EMPLOYER NICKNAME       CHAR            8
EMPLOYER ORIGINAL_DATE  DATE
EMPLOYER PHONE          CHAR           14
EMPLOYER STATE          CHAR            2
EMPLOYER STATUS         CHAR            8
EMPLOYER TITLE          CHAR           36
EMPLOYER TYPE1          CHAR            6
EMPLOYER TYPE2          CHAR            6
EMPLOYER ZIP            CHAR           10
ELO2121I ******** End-of-Data **********
```

```
SELECT TNAME, CNAME, COLTYPE, LENGTH FROM SYSTEM.SYSCOLUMNS WHERE CREATOR = 'SPETER' AND TNAME = 'STUDENT'
TNAME   CNAME               COLTYPE  LENGTH
------- ------------------- -------- -------
STUDENT ACTVY1              CHAR          40
STUDENT ACTVY2              CHAR          40
STUDENT ACTVY3              CHAR          40
STUDENT ACTVY4              CHAR          40
STUDENT ACTVY5              CHAR          40
STUDENT ACTVY6              CHAR          40
STUDENT ACT1_LDR_TITLE      CHAR          15
STUDENT ACT2_LDR_TITLE      CHAR          15
STUDENT ACT3_LDR_TITLE      CHAR          15
STUDENT ACT4_LDR_TITLE      CHAR          15
STUDENT ACT5_LDR_TITLE      CHAR          15
STUDENT ACT6_LDR_TITLE      CHAR          15
STUDENT BID_POINTS_AVAIL    INTEGER
STUDENT CAREER1_INTEREST    CHAR           3
STUDENT CAREER2_INTEREST    CHAR           3
STUDENT CAREER3_INTEREST    CHAR           3
STUDENT CITIZENSHIP         CHAR           1
STUDENT CMP1_SKILL          CHAR          12
STUDENT CMP2_SKILL          CHAR          12
STUDENT CMP3_SKILL          CHAR          12
STUDENT CMP4_SKILL          CHAR          12
STUDENT CMP5_SKILL          CHAR          12
STUDENT CMP6_SKILL          CHAR          12
STUDENT CONCENTRATION       CHAR          20
STUDENT DATE_AVAIL          DATE
STUDENT DEGREE              CHAR           1
STUDENT DIVISION            CHAR           1
STUDENT EDIT_DATE           DATE
STUDENT EDIT_TIME           TIME
STUDENT EMP1_CITY           CHAR          18
STUDENT EMP1_FROM           CHAR           5
STUDENT EMP1_HRS_PER_WEEK   CHAR           2
STUDENT EMP1_JOB_CATEGORY   CHAR           3
STUDENT EMP1_LOCATION       CHAR          18
STUDENT EMP1_MOS_EXPERIENC  CHAR           2
STUDENT EMP1_NAME           CHAR          31
STUDENT EMP1_POSITION       CHAR          80
STUDENT EMP1_STATE          CHAR           2
STUDENT EMP1_TO             CHAR           5
STUDENT EMP2_CITY           CHAR          18
STUDENT EMP2_FROM           CHAR           5
STUDENT EMP2_HRS_PER_WEEK   CHAR           2
STUDENT EMP2_JOB_CATEGORY   CHAR           3
STUDENT EMP2_LOCATION       CHAR          18
STUDENT EMP2_MOS_EXPERIENC  CHAR           2
STUDENT EMP2_NAME           CHAR          31
STUDENT EMP2_POSITION       CHAR          80
STUDENT EMP2_STATE          CHAR           2
STUDENT EMP2_TO             CHAR           5
STUDENT EMP3_CITY           CHAR          18
STUDENT EMP3_FROM           CHAR           5
STUDENT EMP3_HRS_PER_WEEK   CHAR           2
STUDENT EMP3_JOB_CATEGORY   CHAR           3
STUDENT EMP3_LOCATION       CHAR          18
STUDENT EMP3_MOS_EXPERIENC  CHAR           2
STUDENT EMP3_NAME           CHAR          31
STUDENT EMP3_POSITION       CHAR          80
STUDENT EMP3_STATE          CHAR           2
STUDENT EMP3_TO             CHAR           5
STUDENT EMP4_CITY           CHAR          18
STUDENT EMP4_FROM           CHAR           5
STUDENT EMP4_HRS_PER_WEEK   CHAR           2
STUDENT EMP4_JOB_CATEGORY   CHAR           3
STUDENT EMP4_LOCATION       CHAR          18
STUDENT EMP4_MOS_EXPERIENC  CHAR           2
STUDENT EMP4_NAME           CHAR          31
STUDENT EMP4_POSITION       CHAR          80
STUDENT EMP4_STATE          CHAR           2
STUDENT EMP4_TO             CHAR           5
STUDENT FIRST_NAME          CHAR          15
STUDENT GEO_PREF            CHAR           1
STUDENT GEOGRAPH_PREF       CHAR          10
STUDENT GEO1_RESTRICT       CHAR           3
STUDENT GEO2_RESTRICT       CHAR           3
STUDENT GEO3_RESTRICT       CHAR           3
STUDENT GPA_FRESHMAN        DECIMAL  ( 3, 2)
STUDENT GPA_GRADUATE        DECIMAL  ( 3, 2)
STUDENT GPA_JUNIOR          DECIMAL  ( 3, 2)
STUDENT GPA_OVERALL         DECIMAL  ( 3, 2)
STUDENT GPA_SENIOR          DECIMAL  ( 3, 2)
STUDENT GPA_SOPHOMORE       DECIMAL  ( 3, 2)
STUDENT GRAD_MONTH          CHAR           1
STUDENT GRAD_YR             CHAR           2
STUDENT LAST_NAME           CHAR          20
STUDENT M_INITIAL           CHAR           1
STUDENT MAJOR1              CHAR           3
STUDENT MAJOR2              CHAR           3
STUDENT MAJOR3              CHAR           3
STUDENT MINOR               CHAR           3
STUDENT MJR1GPA             DECIMAL  ( 3, 2)
STUDENT MJR2GPA             DECIMAL  ( 3, 2)
STUDENT MJR3GPA             DECIMAL  ( 3, 2)
STUDENT OD1_DEGREE          CHAR           1
STUDENT OD1_GPA             DECIMAL  ( 3, 2)
STUDENT OD1_GRAD_YEAR       CHAR           2
STUDENT OD1_OTHER           CHAR          50
```

```
STUDENT OD1_SCHOOL          CHAR          21
STUDENT OD1MJR1             CHAR           3
STUDENT OD1MJR2             CHAR           3
STUDENT OD2_DEGREE          CHAR           1
STUDENT TEACH4_CERT         CHAR           3
STUDENT TEACH3_CERT         CHAR           3
STUDENT TEACH2_CERT         CHAR           3
STUDENT TEACH1_CERT         CHAR           3
STUDENT TCH2_SUBJECTS       CHAR          18
STUDENT TCH2_STATE          CHAR           2
STUDENT TCH2_SCHOOL_DIST    CHAR          20
STUDENT TCH2_GRADE_LEVELS   CHAR           7
STUDENT TCH2_DATES          CHAR          11
STUDENT TCH2_CITY           CHAR          20
STUDENT TCH1_SUBJECTS       CHAR          18
STUDENT TCH1_STATE          CHAR           2
STUDENT TCH1_SCHOOL_DIST    CHAR          20
STUDENT TCH1_GRADE_LEVELS   CHAR           7
STUDENT TCH1_DATES          CHAR          11
STUDENT TCH1_CITY           CHAR          20
STUDENT STATUS              CHAR           8
STUDENT SSN                 CHAR           9
STUDENT SPECIAL_SKILLS      CHAR          80
STUDENT SIGNATURE_ON_FILE   CHAR           1
STUDENT SEX                 CHAR           1
STUDENT RACE                CHAR           1
STUDENT PRES_ZIP            CHAR           5
STUDENT PRES_STATE          CHAR           2
STUDENT PRES_PHONE          CHAR          14
STUDENT PRES_CITY           CHAR          20
STUDENT PRES_ADDR           CHAR          30
STUDENT PLACEMENT_STATUS    CHAR           1
STUDENT PERM_ZIP            CHAR           5
STUDENT PERM_STATE          CHAR           2
STUDENT PERM_PHONE          CHAR          14
STUDENT PERM_CITY           CHAR          20
STUDENT PERM_ADDR           CHAR          30
STUDENT PERCENT_EARNED      SMALLINT
STUDENT OD2MJR2             CHAR           3
STUDENT OD2MJR1             CHAR           3
STUDENT OD2_SCHOOL          CHAR          21
STUDENT OD2_OTHER           CHAR          50
STUDENT OD2_GRAD_YEAR       CHAR           2
STUDENT OD2_GPA             DECIMAL  ( 3, 2)
EL02121I ******** End-of-Data **********
```

END OF FIGURE 16.

Schemas for the CPPO Relational Database.

BEGINNING OF FIGURE 17.

```
****************************************************************
CERTAIN RELATIONS BEFORE RUNNING OF CPPO0029
****************************************************************

SELECT * FROM BID
SSN       EMP_NO SCH_NO POINTS_BID STATUS   TIME_OF_BID DATE_OF_BID
--------- ------ ------ ---------- -------- ----------- -----------
222222222   999   1          900 UNPROCED 09.08.00     1993-01-12
222222223   999   1          900 UNPROCED 09.08.00     1993-01-13
222222224   999   1          900 UNPROCED 09.08.01     1993-01-12
222222225   999   1          900 UNPROCED 09.08.02     1993-01-12
333332222   999   1          901 UNPROCED 09.01.00     1993-01-12
333333333   999   1          902 UNPROCED 09.02.00     1993-01-12
333334444   999   1          903 UNPROCED 09.03.00     1993-01-12
333335555   999   1          904 UNPROCED 09.04.00     1993-01-12
333336666   999   1          905 UNPROCED 09.04.00     1993-01-12
333337777   999   1          906 UNPROCED 09.05.00     1993-01-12
333338888   999   1          907 UNPROCED 09.06.00     1993-01-12
333339999   999   1          908 UNPROCED 09.07.00     1993-01-12
333331111   999   1          909 UNPROCED 09.00.00     1993-01-12
333331010   999   1          910 UNPROCED 09.09.00     1993-01-12
333331212   999   1          911 UNPROCED 09.10.00     1993-01-12
333331313   999   1          912 UNPROCED 09.11.00     1993-01-12
333331414   999   1          913 UNPROCED 09.14.00     1993-01-12
333331515   999   1          914 UNPROCED 09.15.00     1993-01-12
295364288   999   1          815 UNPROCED 08.50.00     1993-01-12
333333334   999   1          816 UNPROCED 08.55.00     1993-01-12
ELO2121I ******** End-of-Data **********


SELECT EMP_NO, SCH_NO, STATUS, POINTS_USED_TO_WIN FROM VISIT
EMP_NO SCH_NO STATUS   POINTS_USED_TO_WIN
------ ------ -------- ------------------
  999   1     BID                       0
  999   2     CLOSED                    0
  001   1     BID                       0
  005   1     BID                       0
  010   1     BID                       0
  015   1     BID                       0
ELO2121I ******** End-of-Data **********


SELECT SSN, LAST_NAME, BID_POINTS_AVAIL FROM STUDENT
SSN       LAST_NAME            BID_POINTS_AVAIL
--------- -------------------- ----------------
222222222 BERK                             5000
222222223 ALMOST1                          4000
222222224 ALMOST2                          4000
222222225 ALMOST3                          4000
333332222 JONES2                           4000
333333333 JONES3                           4000
333334444 JONES4                           4000
333335555 JONES5                           4000
333336666 JONES6                           4000
333337777 JONES7                           4000
333338888 JONES8                           4000
333339999 JONES9                           4000
333331111 JONES1                           4000
333331212 JONES12                          4000
333331313 JONES13                          4000
333331414 JONES14                          4000
333331515 JONES15                          4000
295364288 SMITH                            4000
333333334 HEIKE                            4000
ELO2121I ******** End-of-Data **********


****************************************************************
RESULTS OF VARIOUS RELATIONS AFTER THE RUNNING OF CPPO0029
****************************************************************

SELECT * FROM BID
SSN       EMP_NO SCH_NO POINTS_BID STATUS   TIME_OF_BID DATE_OF_BID
--------- ------ ------ ---------- -------- ----------- -----------
222222222   999   1          900 WON      09.08.00     1993-01-12
222222223   999   1          900 LOST     09.08.00     1993-01-13
222222224   999   1          900 LOST     09.08.01     1993-01-12
222222225   999   1          900 LOST     09.08.02     1993-01-12
333332222   999   1          901 WON      09.01.00     1993-01-12
333333333   999   1          902 WON      09.02.00     1993-01-12
333334444   999   1          903 WON      09.03.00     1993-01-12
333335555   999   1          904 WON      09.04.00     1993-01-12
333336666   999   1          905 WON      09.04.00     1993-01-12
333337777   999   1          906 WON      09.05.00     1993-01-12
333338888   999   1          907 WON      09.06.00     1993-01-12
333339999   999   1          908 WON      09.07.00     1993-01-12
333331111   999   1          909 WON      09.00.00     1993-01-12
333331010   999   1          910 WON      09.09.00     1993-01-12
333331212   999   1          911 WON      09.10.00     1993-01-12
333331313   999   1          912 WON      09.11.00     1993-01-12
333331414   999   1          913 WON      09.14.00     1993-01-12
333331515   999   1          914 WON      09.15.00     1993-01-12
295364288   999   1          815 LOST     08.50.00     1993-01-12
333333334   999   1          816 LOST     08.55.00     1993-01-12
ELO2121I ******** End-of-Data **********


SELECT EMP_NO, SCH_NO, STATUS, POINTS_USED_TO_WIN FROM VISIT
EMP_NO SCH_NO STATUS   POINTS_USED_TO_WIN
------ ------ -------- ------------------
  999   1     #BD>=SLT                900
  999   2     CLOSED                    0
  001   1     BID                       0
  005   1     BID                       0
  010   1     BID                       0
  015   1     BID                       0
ELO2121I ******** End-of-Data **********


SELECT SSN, LAST_NAME, BID_POINTS_AVAIL FROM STUDENT
SSN       LAST_NAME            BID_POINTS_AVAIL
--------- -------------------- ----------------
222222222 BERK                             5000
222222223 ALMOST1                          4900
```

```
222222224 ALMOST2                          4900
222222225 ALMOST3                          4900
333332222 JONES2                           4001
333333333 JONES3                           4002
333334444 JONES4                           4003
333335555 JONES5                           4004
333336666 JONES6                           4005
333337777 JONES7                           4006
333338888 JONES8                           4007
333339999 JONES9                           4008
333331111 JONES1                           4009
333331212 JONES12                          4011
333331313 JONES13                          4012
333331414 JONES14                          4013
333331515 JONES15                          4014
295364288 SMITH                            4815
333333334 HEIKE                            4816
KLO2121I ******** End-of-Data **********
```

END OF FIGURE 17.

Before and After Bids Are Processed.

c. Program Source Code

1) Source Code of Module: CPPOCICS

```
*                                                                CIC00010
*****************************************************************CIC00020
*  CPPOCICS - Test Backbone CPPO programs with pseudo CICS.     *CIC00030
*****************************************************************CIC00040
                                                                 CIC00050
 IDENTIFICATION DIVISION.                                        CIC00060
 PROGRAM-ID.  CPPOCICS.                                            CIC00070
 AUTHOR.      T. SCHABER.                                        CIC00080
                                                                 CIC00090
* PROGRAM FUNCTIONAL REQUIREMENTS                                CIC00100
* Driver program to initiate execution of one of several         CIC00110
* backbone programs for the CPPO system. This program simulates  CIC00120
* CICS in that it calls the requested program and passes the     CIC00130
* DFHCOMMAREA to the called program.                             CIC00140
                                                                 CIC00150
 eject                                                           CIC00160
 ENVIRONMENT DIVISION.                                           CIC00170
 CONFIGURATION SECTION.                                          CIC00180
 SPECIAL-NAMES.                                                  CIC00190
     C01 IS TOP-OF-PAGE.                                         CIC00200
                                                                 CIC00210
 INPUT-OUTPUT SECTION.                                           CIC00220
 FILE-CONTROL.                                                   CIC00230
                                                                 CIC00240
                                                                 CIC00250
 DATA DIVISION.                                                  CIC00260
 FILE SECTION.                                                   CIC00270
                                                                 CIC00280
 WORKING-STORAGE SECTION.                                        CIC00290
     Copy DFHCOMM.                                               CIC00300
 01  DFHCOMMAREA-200-BYTES REDEFINES DFHCOMMAREA.                CIC00310
     05  FIRST-50-BYTES          PIC X(50).                     CIC00320
     05  SECOND-50-BYTES         PIC X(50).                     CIC00330
     05  THIRD-50-BYTES          PIC X(50).                     CIC00340
     05  FOURTH-50-BYTES         PIC X(50).                     CIC00350
                                                                 CIC00360
 eject                                                           CIC00370
*****************************************************************CIC00380
*    PROGRAM VARIABLE DECLARTION SECTION                        *CIC00390
*****************************************************************CIC00400
 01  STEP-INDICATOR          PIC X(30) VALUE 'PREP-SCREEN'.     CIC00410
 01  DB-STATUS-INDICATOR     PIC X(8)  VALUE SPACES.            CIC00420
 01  EDIT-STATUS-INDICATOR   PIC X(8)  VALUE SPACES.            CIC00430
 01  THIS-PROGRAM-ID         PIC X(8)  VALUE 'CPPO'.            CIC00440
 01  CALL-PROGRAM-ID         PIC X(8).                          CIC00450
 01  SCRN-NAME               PIC X(8)  VALUE 'CICSSCRN'.        CIC00460
 01  SCRN-RETCODE            PIC S9(7) COMP.                    CIC00470
 01  SCRN-NUMBER             PIC S9(7) COMP.                    CIC00480
 01  SCRN-FIELDS.                                               CIC00500
     05 COMM1           PIC    X(50).                         CIC00510
     05 COMM2           PIC    X(50).                         CIC00520
     05 COMM3           PIC    X(50).                         CIC00530
     05 COMM4           PIC    X(50).                         CIC00550
 01 SCRN-KEY   PIC      X(8).                                  CIC00560
     88  PF1                          VALUE 'PF01'.            CIC00570
     88  PF2                          VALUE 'PF02'.            CIC00580
     88  PF3                          VALUE 'PF03'.            CIC00590
     88  PF4                          VALUE 'PF04'.            CIC00600
     88  PF5                          VALUE 'PF05'.            CIC00610
     88  PF6                          VALUE 'PF06'.            CIC00620
     88  PF7                          VALUE 'PF07'.            CIC00630
     88  PF8                          VALUE 'PF08'.            CIC00640
     88  PF9                          VALUE 'PF09'.            CIC00650
     88  PF10                         VALUE 'PF10'.            CIC00660
     88  PF11                         VALUE 'PF11'.            CIC00670
     88  PF12                         VALUE 'PF12'.            CIC00680
     88  RETURN-KEY                   VALUE 'RETURN'.
```

```
                                                                 CIC00690
 eject                                                           CIC00700
 PROCEDURE DIVISION.                                             CIC00710
 0000-MAINLINE SECTION.                                          CIC00720
     PERFORM 0000-INITIALIZE-PROGRAM.                            CIC00730
     PERFORM UNTIL STEP-INDICATOR = 'STOP-PROGRAM'               CIC00740
        EVALUATE TRUE                                            CIC00750
            WHEN STEP-INDICATOR = 'PREP-SCREEN'                  CIC00760
                 PERFORM 1000-PREPARE-SCREEN-OUTPUT              CIC00770
            WHEN STEP-INDICATOR = 'DISP-SCREEN'                  CIC00780
                 PERFORM 2000-DISPLAY-SCREEN                     CIC00790
            WHEN STEP-INDICATOR = 'EVAL-PFKEYS'                  CIC00800
                 PERFORM 3000-EVALUATE-PFKEYS                    CIC00810
            WHEN STEP-INDICATOR = 'XFER-CONTROL'                 CIC00820
                 PERFORM 4000-TRANSFER-CONTROL                   CIC00830
            WHEN OTHER                                           CIC00840
               DISPLAY 'STEP INDICATOR NOT SET'                  CIC00850
        END-EVALUATE                                             CIC00860
     END-PERFORM.                                                CIC00870
     CALL 'MEXIT' USING SCRN-NUMBER, SCRN-RETCODE.               CIC00880
     STOP RUN.                                                   CIC00900
 EXIT-MAINLINE.                                                  CIC00910
 eject                                                           CIC00920
 0000-INITIALIZE-PROGRAM.                                        CIC00930
     MOVE ALL '1234567890' TO DFHCOMMAREA.                       CIC00990
     MOVE 'CICS     ' TO NEXT-PROGRAM-ID.                        CIC00940
     MOVE '222222222' TO COMM-STUDENT-SSN.                       CIC00950
     CALL 'MLOAD' USING SCRN-NUMBER, SCRN-RETCODE, SCRN-NAME.    CIC00970
*    Replace with data you want in the DFHCOMMAREA when your     CIC00980
*    program gets control.                                       CIC01000
     MOVE 'PREP-SCREEN' TO STEP-INDICATOR.                       CIC01010
                                                                 CIC01020
 1000-PREPARE-SCREEN-OUTPUT SECTION.                             CIC01030
     MOVE  FIRST-50-BYTES   TO COMM1.                            CIC01040
     MOVE  SECOND-50-BYTES  TO COMM2.                            CIC01050
     MOVE  THIRD-50-BYTES   TO COMM3.                            CIC01060
     MOVE  FOURTH-50-BYTES  TO COMM4.                            CIC01070
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.                       CIC01080
 EXIT-PREPARE-SCREEN-OUTPUT.                                     CIC01090
 eject                                                           CIC01100
                                                                 CIC01110
 2000-DISPLAY-SCREEN SECTION.                                    CIC01120
     CALL 'CICSSCRN' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.   CIC01130
     MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.                       CIC01140
     MOVE  COMM1 TO FIRST-50-BYTES.                              CIC01150
     MOVE  COMM2 TO SECOND-50-BYTES.                             CIC01160
     MOVE  COMM3 TO THIRD-50-BYTES.                              CIC01170
     MOVE  COMM4 TO FOURTH-50-BYTES.                             CIC01180
 EXIT-DISPLAY-SCREEN.                                            CIC01190
 eject                                                           CIC01200
 3000-EVALUATE-PFKEYS SECTION.                                   CIC01210
*    Replace CONTINUE statements with appropriate application code CIC01220
     EVALUATE TRUE                                               CIC01230
        WHEN RETURN-KEY                                          CIC01240
             CONTINUE                                            CIC01250
*      WHEN PF1                                                   CIC01260
*           CALL 'CPPO0038' USING DFHCOMMAREA                     CIC01270
*       WHEN PF2                                                  CIC01280
*            CALL 'CPPOEXIT' USING DFHCOMMAREA                    CIC01290
        WHEN PF3                                                 CIC01300
             MOVE 'STOP-PROGRAM' TO STEP-INDICATOR               CIC01310
             GO TO EXIT-EVALUATE-PFKEYS                          CIC01320
        WHEN PF4                                                 CIC01330
             MOVE 'CPPO0010' TO CALL-PROGRAM-ID
             CALL CALL-PROGRAM-ID USING DFHCOMMAREA
             CANCEL CALL-PROGRAM-ID
```

```
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR                CIC01340
                GO TO EXIT-EVALUATE-PFKEYS                           CIC01360
                                                                     CIC01370
            WHEN PF5                                                 CIC01380
      *         CALL 'CPPO0011' USING DFHCOMMAREA
                DISPLAY 'TO GET TO THIS OPTION'
                DISPLAY 'MUST PRESS PF4 AND THEN SELECT A COMPANY'
      *         MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01390
      *         GO TO EXIT-EVALUATE-PFKEYS                         CIC01400
                                                                                CIC01410
            WHEN PF6
                MOVE 'CPPO0015' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR

CIC01340
                GO TO EXIT-EVALUATE-PFKEYS                         CIC01360
            WHEN PF7                                               CIC01430
                MOVE 'CPPO0014' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01340
                GO TO EXIT-EVALUATE-PFKEYS                         CIC01360
      *       WHEN PF8                                             CIC01450
      *         CALL 'CPPO0020' USING DFHCOMMAREA                  CIC01460
            WHEN PF9                                               CIC01470
                MOVE 'CPPO0029' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'DISP-SCREEN' TO STEP-INDICATOR               CIC01340
                GO TO EXIT-EVALUATE-PFKEYS                         CIC01360
      *       WHEN PF10                                            CIC01490
      *         CALL 'SUBR2' USING DFHCOMMAREA                     CIC01500
            WHEN OTHER                                             CIC01510
                DISPLAY 'INVALID KEY WAS PRESSED'                  CIC01520
        END-EVALUATE.                                              CIC01530
        MOVE 'PREP-SCREEN' TO STEP-INDICATOR.                      CIC01540
     EXIT-EVALUATE-PFKEYS.                                         CIC01550
     eject                                                         CIC01560
     4000-TRANSFER-CONTROL SECTION.                                CIC01570
        IF NEXT-PROGRAM-ID = 'CICS    ' THEN
          MOVE 'PREP-SCREEN' TO STEP-INDICATOR                     CIC01120
        ELSE
        IF NEXT-PROGRAM-ID = 'CPPO0010' THEN
                MOVE 'CPPO0010' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01120
        ELSE
        IF NEXT-PROGRAM-ID = 'CPPO0011' THEN
                MOVE 'CPPO0011' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01120
        ELSE
        IF NEXT-PROGRAM-ID = 'CPPO0014' THEN
                MOVE 'CPPO0014' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01120
        ELSE
        IF NEXT-PROGRAM-ID = 'CPPO0015' THEN
                MOVE 'CPPO0015' TO CALL-PROGRAM-ID
                CALL CALL-PROGRAM-ID USING DFHCOMMAREA             CIC01330
                CANCEL CALL-PROGRAM-ID
```

```
                MOVE 'XFER-CONTROL' TO STEP-INDICATOR              CIC01120
        ELSE
        MOVE 'PREP-SCREEN' TO STEP-INDICATOR.                      CIC01120

     EXIT-TRANSFER-CONTROL.                                        CIC01580
     *EJECT                                                        CIC01590
                                                                   CIC01600
```

2) Source Code of Module: CPPO0010

```
*
*****************************************************************
*  CPPO0010  -  COMPANIES INTERVIEWING FOR MAJOR.
*****************************************************************

 IDENTIFICATION DIVISION.
 PROGRAM-ID.   CPPO0010.
 AUTHOR.       ROGER TATE.

*MAJOR MODIFICATIONS 10/92 STEVE PETER
*  CHANGES THAT NEED TO BE DONE:
*     GET THE "LIKE" FEATURE TO WORK WITH HOST VARIBLE IN QUERY.
*     MAKE SURE THE QUERY ON VISIT IS IN ORDER BY DATE
*         SO STUDENT SEES LIST IN ORDER BY DATE
*                                        11/12/92 S.PETER.
*
*
* PROGRAM FUNCTIONAL REQUIREMENTS:
*
* DISPLAYS COMPANIES THAT THE STUDENT IS ELIBIBLE TO BID ON
* BASED ON THEIR DEGREE, MAJOR, GRADUATION DATE
* AND CLOSED STATUS.  ALLOWS USER TO SELECT
* THE EMPLOYER THAT THEY DESIRE TO BID FOR OR TO JUST SEE
* ADDITIONAL INFORMATION ABOUT THE COMPANY.  RECEIVES
* STUDENT SSN FROM CPPO0001 AND PASSES COMPANY # AND
* SCHEDULE # SELECTED BY THE USER TO CPPO0011.
*
 eject
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
     C01 IS TOP-OF-PAGE.

 INPUT-OUTPUT SECTION.
 FILE-CONTROL.


 DATA DIVISION.
 FILE SECTION.

 WORKING-STORAGE SECTION.

     EXEC SQL BEGIN DECLARE SECTION END-EXEC.                    0077000
*    FOR EMPLOYER TABLE
 01 SQL-EMPLOYER-NAME                    PIC X(40).              0097000

*    FOR STUDENT TABLE
 01  SQL-STU-SSN                        PIC X(09).              TU00040
 01  SQL-STU-LAST-NAME                  PIC X(20).              TU00050
 01  SQL-STU-FIRST-NAME                 PIC X(15).              TU00060
 01  SQL-STU-M-INITIAL                  PIC X(01).              TU00070
                                                                TU00080
*    ADDRESS                                                    TU00090
 01  SQL-STU-PRES-ADDR                  PIC X(20).              TU00100
 01  SQL-STU-PRES-CITY                  PIC X(20).              TU00110
 01  SQL-STU-PRES-STATE                 PIC X(02).              TU00120
 01  SQL-STU-PRES-ZIP                   PIC X(05).              TU00130
 01  SQL-STU-PRES-PHONE                 PIC X(10).              TU00140
 01  SQL-STU-PERM-ADDR                  PIC X(20).              TU00150
 01  SQL-STU-PERM-CITY                  PIC X(20).              TU00160
 01  SQL-STU-PERM-STATE                 PIC X(02).              TU00170
 01  SQL-STU-PERM-ZIP                   PIC X(05).              TU00180
 01  SQL-STU-PERM-PHONE                 PIC X(10).              TU00190
                                                                TU00200
*    DEMOGRAPHICS                                               TU00210
 01  SQL-STU-SEX                        PIC X(01).              TU00220

 01  SQL-STU-CITIZENSHIP                PIC X(01).              TU00230
                                                                TU00240
*    CAREER INTEREST DATA                                       TU00250
 01  SQL-STU-CAREER-INT                 PIC X(11).              TU00260
 01  SQL-STU-GEOGRAPH-PREF              PIC X(10).              TU00270
 01  SQL-STU-WORK-PREF                  PIC X(20).              TU00280
 01  SQL-STU-DATE-AVAIL                 PIC X(10).              TU00290
                                                                TU00300
*    ACADEMIC RECORD                                            TU00310
 01  SQL-STU-DIVISION                   PIC X(01).              TU00320
 01  SQL-STU-MAJOR-1                    PIC X(03).              TU00330
 01  SQL-STU-MAJOR-2                    PIC X(03).              TU00340
 01  SQL-STU-MAJOR-3                    PIC X(03).              TU00340
 01  SQL-STU-MINOR                      PIC X(03).              TU00350
 01  SQL-STU-CONCENTRATION              PIC X(20).              TU00360
 01  SQL-STU-DEGREE                     PIC X(01).              TU00370
 01  SQL-STU-GRAD-DATE                  PIC X(03).
 01  SQL-STU-GRAD-MONTH                 PIC X(01).              TU00380
 01  SQL-STU-GRAD-YR                    PIC X(02).              TU00380
 77  SQL-STU-GPA-OVERALL                PIC S9V99 COMP-3.       TU00390
 77  SQL-STU-GPA-MAJOR-1                PIC S9V99 COMP-3.       TU00400
 77  SQL-STU-GPA-MAJOR-2                PIC S9V99 COMP-3.       TU00410
 77  SQL-STU-GPA-FRESHMAN               PIC S9V99 COMP-3.       TU00420
 77  SQL-STU-GPA-SOPHOMORE              PIC S9V99 COMP-3.       TU00430
 77  SQL-STU-GPA-JUNIOR                 PIC S9V99 COMP-3.       TU00440
 77  SQL-STU-GPA-SENIOR                 PIC S9V99 COMP-3.       TU00450
 77  SQL-STU-GPA-GRADUATE               PIC S9V99 COMP-3.       TU00460
                                                                TU00470
*    CPPO DATA                                                  TU00480
 77  SQL-STU-BID-POINTS-AVAIL           PIC S9(04) COMP.        TU00490
                                                                TU00500
*    EMPLOYMENT HISTORY                                         TU00510
 01  SQL-STU-PERCENT-EARNED             PIC S9(4) COMP.         TU00520
 01  SQL-STU-EMPLOYER-1                 PIC X(20).              TU00530
 01  SQL-STU-LOCATION-1                 PIC X(20).              TU00540
 01  SQL-STU-JOB-CATAGORY-1             PIC X(03).              TU00550
 01  SQL-STU-FROM-DATE-1                PIC X(04).              TU00560
 01  SQL-STU-TO-DATE-1                  PIC X(04).              TU00570
 01  SQL-STU-WORK-DESC-1                PIC X(40).              TU00580
 01  SQL-STU-EMPLOYER-2                 PIC X(20).              TU00590
 01  SQL-STU-LOCATION-2                 PIC X(20).              TU00600
 01  SQL-STU-JOB-CATAGORY-2             PIC X(03).              TU00610
 01  SQL-STU-FROM-DATE-2                PIC X(04).              TU00620
 01  SQL-STU-TO-DATE-2                  PIC X(04).              TU00630
 01  SQL-STU-WORK-DESC-2                PIC X(40).              TU00640
 01  SQL-STU-EMPLOYER-3                 PIC X(20).              TU00650
 01  SQL-STU-LOCATION-3                 PIC X(20).              TU00660
 01  SQL-STU-JOB-CATAGORY-3             PIC X(03).              TU00670
 01  SQL-STU-FROM-DATE-3                PIC X(04).              TU00680
 01  SQL-STU-TO-DATE-3                  PIC X(04).              TU00690
 01  SQL-STU-WORK-DESC-3                PIC X(40).              TU00700
                                                                TU00710
*    ACTIVITIES, HONORS,SKILLS                                  TU00720
 01  SQL-STU-ACTIVITIES                 PIC X(240).             TU00730
 01  SQL-STU-SPECIAL-SKILLS             PIC X(40).              TU00740
                                                                TU00750
*    TEACHING DATA                                              TU00760
 01  SQL-STU-TEACHER-CERT-1             PIC X(40).              TU00770
 01  SQL-STU-TEACHER-CERT-2             PIC X(40).              TU00780
 01  SQL-STU-INTERNSHIP                 PIC X(40).              TU00790
                                                                TU00800
*    CPPO DATA                                                  TU00810
 01  SQL-STU-SUSPEND-STATUS             PIC X(01).              TU00820
                                                                TU00830
*    COLLEGES ATTENDED                                          TU00840
 01  SQL-STU-OTHER-COLLEGE-1            PIC X(254).             TU00850
```

```
 01  SQL-STU-OTHER-COLLEGE-2            PIC X(254).              TU00860
                                                                  TU00880
 eject                                                            TU00890
*    FOR SCHEDULE TABLE
 01  SQL-SCHEDUL-EMP-NO                 PIC X(5).
 01  SQL-SCHEDUL-SCH-NO                 PIC X(2).
 01  SQL-SCHEDUL-INT-DATE1              PIC X(10).
 01  SQL-SCHEDUL-INT-DATE2              PIC X(10).
 01  SQL-SCHEDUL-INT-DATE3              PIC X(10).
 01  SQL-SCHEDUL-NO-SLOTS               PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-DAYS                PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-ROOMS               PIC S9(4) COMP.
 01  SQL-SCHEDUL-INT-LENGTH             PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-INTERVIEWERS        PIC S9(4) COMP.
 01  SQL-SCHEDUL-SCHED-STATUS           PIC X(6).
 01  SQL-SCHEDUL-DEGREE-REQMT           PIC X(3).
 01  SQL-SCHEDUL-MAJOR-REQMT            PIC X(48).
 01  SQL-SCHEDUL-GPA-MINIMUM            PIC S9V99 COMP-3.
 01  SQL-SCHEDUL-POSITION-AVAIL         PIC X(48).
 01  SQL-SCHEDUL-JOB-LOCATION           PIC X(34).
 01  SQL-SCHEDUL-CITIZENSHIP            PIC X.
 01  SQL-SCHEDUL-GRAD-DATE              PIC X(3).
 01  SQL-SCHEDUL-GRAD-MONTH             PIC X.
 01  SQL-SCHEDUL-GRAD-YEAR              PIC XXX.
 01  SQL-SCHEDUL-LITERATURE             PIC X.
 01  SQL-SCHEDUL-VIDEO                  PIC X.
 01  SQL-SCHEDUL-PRE-NIGHT-LOCATION     PIC X(20).
 01  SQL-SCHEDUL-PRE-NIGHT-DATE         PIC X(10).
 01  SQL-SCHEDUL-PRE-NIGHT-TIME         PIC X(10).
 01  SQL-SCHEDUL-BID-DATE               PIC X(10).
 01  SQL-SCHEDUL-NOTES                  PIC X(60).
 01  SQL-SCHEDUL-POINTS-USED-TO-WIN     PIC S9(4) COMP.

 01  LINK-SSN              PIC X(9).
 01  HOLD-EMP-NO           PIC X(5).
 01  HOST-BID-DATE         PIC X(10).
 01  HOST-DATE.
     05  FILLER            PIC X(02)  VALUE '19'.
     05  HOST-YEAR         PIC X(02).
     05  FILLER            PIC X(01)  VALUE '-'.
     05  HOST-MONTH        PIC X(02).
     05  FILLER            PIC X(01)  VALUE '-'.
     05  HOST-DAY          PIC X(02).
 77  NI                    PIC S9(4) COMP.

     EXEC SQL END DECLARE SECTION END-EXEC.


  01  PREPARED-SCRNDATE.
     10 SCRNDATE-MONTH          PIC X(2).
     10 FILLER                  PIC X(1)  VALUE '/'.
     10 SCRNDATE-DAY            PIC X(2).
     10 FILLER                  PIC X(1)  VALUE '/'.
     10 SCRNDATE-YEAR           PIC X(2).
  01 PREPARED-SCRNTIME.
     10 SCRNTIME-HOUR           PIC X(2).
     10 FILLER                  PIC X(1)  VALUE ':'.
     10 SCRNTIME-MINUTE         PIC X(2).
     10 FILLER                  PIC X(3)  VALUE SPACES.

************************************************************
 01  SCRN-NUMBER    PIC     9(7) COMP.

 01  SCRN-FIELDS.
      05 PGMID          PIC     X(8).
      05 SCRNID         PIC     X(8).
```

```
      05 SCRDATE        PIC     X(8).
      05 SCRTIME        PIC     X(8).
      05 STUMAJ         PIC     X(3).
      05 MORESCR        PIC     X(4).
      05 SLINE1         PIC    X(76).
      05 SLINE2         PIC    X(76).
      05 SLINE3         PIC    X(76).
      05 SLINE4         PIC    X(76).
      05 SLINE5         PIC    X(76).
      05 SLINE6         PIC    X(76).
      05 SLINE7         PIC    X(76).
      05 SLINE8         PIC    X(76).
      05 SLINE9         PIC    X(76).
      05 SENTRY          PIC     X(1).
      05 SYSMSG         PIC    X(72).
 01  SCRN-KEY       PIC     X(8).
         88  PF1                                VALUE 'PF01'.
         88  PF2                                VALUE 'PF02'.
         88  PF3                                VALUE 'PF03'.
         88  PF4                                VALUE 'PF04'.
         88  PF5                                VALUE 'PF05'.
         88  PF6                                VALUE 'PF06'.
         88  PF7                                VALUE 'PF07'.
         88  PF8                                VALUE 'PF08'.
         88  PF9                                VALUE 'PF09'.
         88  PF10                               VALUE 'PF10'.
         88  PF11                               VALUE 'PF11'.
         88  PF12                               VALUE 'PF12'.
         88  RETURN-KEY                         VALUE 'RETURN'.
************************************************************

 eject                                                          00099000
     EXEC SQL INCLUDE SQLCA END-EXEC.                           00100000
*    Additional variables for abnormal termination.
 01  DECODED-SQLCODE        PIC --------999.                    00107000
                                                                00108000
 01  ARRAY-SQLERRD.                                            00109000
     02 DECODED-SQLERRD     PIC --------999 OCCURS 6 TIMES.     00110000
 01  INDX2                  PIC S9(1) SYNC USAGE IS COMP.       00111000
 01  INDXPIC                PIC ZZZ9.                           00112000
 eject                                                          00101000
************************************************************** 00102000
*   PROGRAM VARIABLE DECLARTION SECTION                        * 00103000
************************************************************** 00104000
 01  STEP-INDICATOR         PIC X(30) VALUE 'PREP-SCREEN'.       001050
 01  DB-STATUS-INDICATOR    PIC X(8)  VALUE SPACES.
 01  EDIT-STATUS-INDICATOR  PIC X(8)  VALUE SPACES.
 01  THIS-PROGRAM-ID        PIC X(8)  VALUE 'CPPO0010'.
 01  SCRN-NAME              PIC X(8)  VALUE 'SCRN1120'.
 01  SCRN-RETCODE           PIC S9(7) COMP.
 01  SCRN-FIELD-NAME        PIC X(7).
 01  FIELDNAME              PIC X(7).
 01  EMPLOYER-TABLE.
     05  EMPLOYER-SCHEDULE OCCURS 500 TIMES.
       10  SCHEDUL-EMP-NO                 PIC X(5) VALUE SPACES.
         10  SCHEDUL-SCH-NO                 PIC X(2).
         10  SCHEDUL-EMP-NAME               PIC X(40).
*        10  SCHEDUL-INT-DATE1              PIC X(10).
*        10  SCHEDUL-INT-DATE2              PIC X(10).
*        10  SCHEDUL-INT-DATE3              PIC X(10).
         10  SCHEDUL-DEGREE-REQMT           PIC X(3).
         10  SCHEDUL-MAJOR-REQMT            PIC X(48).
         10  SCHEDUL-sched-status           PIC X(6).
         10  SCHEDUL-POSITION-AVAIL         PIC X(48).
         10  SCHEDUL-JOB-LOCATION           PIC X(34).
*        10  SCHEDUL-GRAD-DATE              PIC XXX.
```

```
           10  SCHEDUL-BID-DATE.
               15  SCHEDUL-BID-DATE-DAY        PIC X(2).
               15  FILLER                      PIC X(1) VALUE '/'.
               15  SCHEDUL-BID-DATE-MONTH      PIC X(2).
*          10  SCHEDUL-NO-SLOTS                PIC S9(4).
*          10  SCHEDUL-NOTES                   PIC X(60).
    01  SCRNOUT-FORMAT-TABLE.
        05  SCRNOUT-FORMAT OCCURS 500 TIMES.
          10 SF-EMPLOYER                        PIC X(25).
          10 FILLER                             PIC X(2).
          10 SF-POSITION                        PIC X(20).
          10 FILLER                             PIC X(1).
          10 SF-LOCATION                        PIC X(15).
          10 FILLER                             PIC X(1).
          10 SF-STATUS                          PIC X(6).
          10 FILLER                             PIC X(1).
          10 SF-BID-DATE                        PIC X(5).

    01  TEMP-SF-BID-DATE.
        05  TEMP-SF-BID-DATE-MONTH              PIC X(2).
        05  FILLER                              PIC X(1) VALUE '/'.
        05  TEMP-SF-BID-DATE-DAY                PIC X(2).

    01  TEMP-BID-DATE.
        05  TEMP-BID-YEAR                       PIC X(4).
        05  FILLER                              PIC X(1).
        05  TEMP-BID-MONTH                      PIC X(2).
        05  FILLER                              PIC X(1).
        05  TEMP-BID-DAY                        PIC X(2).


    01 MISCELLANEOUS.
        05  LINK-EMP-NO           PIC X(5).
        05  LINK-SCH-NO           PIC X(2).
        05  ENTRY-NUMBER          PIC 9.
        05  COUNTER               PIC 999  VALUE 1.
        05  COUNTER-BEFORE        PIC 999.
        05  ROWS-IN-TABLE-TO-COUNT-BACK   PIC 9.
        05  LOOP-NUM              PIC  99  VALUE 1.
        05  PF3-PRESSED           PIC X(1) VALUE 'N'.
        05  LINE-NUMBER           PIC 999.
        05  PAGE-INDICATOR        PIC 999  VALUE 1.
        05  EMP-IDX               PIC 999.
        05  EMPLOYER-NUMBER OCCURS 9 TIMES.
            10 T-EMP-NO           PIC X(5).
            10 T-SCH-NO           PIC X(2).
        05  CURRENT-DATE.
            10 CURRENT-YEAR       PIC X(2).
            10 CURRENT-MONTH      PIC X(2).
            10 CURRENT-DAY        PIC X(2).
        05  CURRENT-TIME.
            10 CURRENT-HOUR       PIC X(2).
            10 CURRENT-MINUTE     PIC X(2).
            10 CURRENT-SECOND     PIC X(2).
            10 CURRENT-HUNDRETH   PIC X(2).

    eject

    * SCREEN ATTRIBUTES
        77 PROT                   PIC X(7) VALUE 'PROT   '.
        77 UNPROT                 PIC X(7) VALUE 'UNPROT '.
        77 NUMRIC                 PIC X(7) VALUE 'NUMERIC'.
        77 BRIGHT                 PIC X(7) VALUE 'BRIGHT '.
        77 DIM                    PIC X(7) VALUE 'DIM    '.
        77 DARK                   PIC X(7) VALUE 'DIM    '.
        77 SKIP                   PIC X(7) VALUE 'SKIP   '.
```

```
        77 NOSKIP                 PIC X(7) VALUE 'NOSKIP '.
        77 MDT                    PIC X(7) VALUE 'MDT    '.
        77 NOMDT                  PIC X(7) VALUE 'NOMDT  '.
    eject                                                              00114000
    01  TERMINAL-MESSAGES.                                            00115000
        02  REC-NOT-FOUND      PIC X(80) VALUE                        00116000
           ' RECORD NOT FOUND - USE A DIFFERENT KEY.'.                 00117000
        02  DUPLICATE-RECORD   PIC X(80) VALUE                        00118000
           ' ATTEMPT TO ADD A DUPLICATE RECORD WAS REJECTED.'.
        02  INVALID-SELECTION  PIC X(80) VALUE
           ' INVALID SELECTION - CHOOSE A NUMBER BETWEEN 1 AND 9.'.
        02  NO-EMPLOYER        PIC X(80) VALUE
           ' THERE IS NO EMPLOYER ON THAT LINE.                    '.
        02  INVALID-FUNCTION-KEY PIC X(80) VALUE
           ' INVALID FUNCTION KEY - CHOOSE ONE THAT IS DISPLAYED.'.
        02  BEGINNING-PAGE     PIC X(80) VALUE
           ' ALREADY AT BEGINNING OF EMPLOYER LIST.'.
        02  ENDING-PAGE        PIC X(80) VALUE
           ' THERE ARE NO MORE EMPLOYER''S TO DISPLAY.'.


    ****************************************************************
    *    We'll use the same name that CICS uses for data passing.
     LINKAGE SECTION.
     COPY DFHCOMM.
    ****************************************************************
                                                                       00106000
     eject
     PROCEDURE DIVISION USING DFHCOMMAREA.
    **    EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
     eject
     0000-MAINLINE.
         CALL 'MLOAD' USING  SCRN-NUMBER, SCRN-RETCODE, SCRN-NAME.
         IF SCRN-RETCODE NOT = 0 THEN
             DISPLAY 'UNABLE TO OBTAIN SCREEN.  EXITING PROGRAM.'
             MOVE 'CPP00001' TO NEXT-PROGRAM-ID
             PERFORM 4000-TRANSFER-CONTROL
             GOBACK.
         ACCEPT CURRENT-DATE FROM DATE.
         MOVE CURRENT-DAY  TO HOST-DAY.
         MOVE CURRENT-MONTH TO HOST-MONTH.
         MOVE CURRENT-YEAR TO HOST-YEAR.
         MOVE HOST-DATE TO HOST-BID-DATE.
         ACCEPT CURRENT-TIME FROM TIME.

         PERFORM 90100-GET-DFHCOMMAREA.
         PERFORM 95100-GET-SELECTED-STUDENT.
         PERFORM 95200-OPEN-SCHD-FOR-BROWSE.
         PERFORM 95300-GET-NEXT-SCHEDULE VARYING EMP-IDX FROM 1 BY 1
             UNTIL EMP-IDX > 500 OR SQLCODE = 100.
         PERFORM UNTIL STEP-INDICATOR = 'XFER-CONTROL'
           EVALUATE TRUE
               WHEN STEP-INDICATOR = 'PREP-SCREEN'
                    PERFORM 1000-PREPARE-SCREEN-OUTPUT
               WHEN STEP-INDICATOR = 'DISP-SCREEN'
                    PERFORM 2000-DISPLAY-SCREEN
               WHEN STEP-INDICATOR = 'EVAL-PFKEYS'
                    PERFORM 3000-EVALUATE-PFKEYS
               WHEN OTHER
    *               program logic error if control reaches this point.
                    DISPLAY 'STEP INDICATOR NOT SET'
                    PERFORM 99000-BACKOUT
           END-EVALUATE
         END-PERFORM.
         MOVE 'PREP-SCREEN' TO STEP-INDICATOR.
```

```
    PERFORM 95500-CLOSE-SCHEDUL-CURSOR.
    PERFORM 4000-TRANSFER-CONTROL.
    CALL 'MPURGE' USING SCRN-NUMBER SCRN-RETCODE.
    GOBACK.
EXIT-MAINLINE.
    EXIT.
eject

1000-PREPARE-SCREEN-OUTPUT.
    PERFORM 1010-INITIALIZE-SCRN-VARIABLES.
    MOVE ZERO TO LINE-NUMBER.

    MOVE CURRENT-DAY                TO SCRNDATE-DAY.
    MOVE CURRENT-MONTH              TO SCRNDATE-MONTH.
    MOVE CURRENT-YEAR               TO SCRNDATE-YEAR.
    MOVE PREPARED-SCRNDATE          TO SCRDATE.

    MOVE CURRENT-HOUR               TO SCRNTIME-HOUR.
    MOVE CURRENT-MINUTE             TO SCRNTIME-MINUTE.
    MOVE PREPARED-SCRNTIME          TO SCRTIME.
    MOVE THIS-PROGRAM-ID            TO PGMID.
    MOVE SCRN-NAME                  TO SCRNID.
    MOVE SQL-STU-MAJOR-1            TO STUMAJ.
    PERFORM 1100-SCREEN-OF-EMPLOYERS UNTIL LOOP-NUM = 10.
    MOVE 1 TO LOOP-NUM.
    IF SCHEDUL-EMP-NO (COUNTER + 1) = '    ' THEN
        MOVE SPACES                 TO MORESCR
    ELSE
        MOVE 'MORE'                 TO MORESCR.
    MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-PREPARE-SCREEN-OUTPUT.

1010-INITIALIZE-SCRN-VARIABLES.
    MOVE SPACES TO SLINE1.
    MOVE SPACES TO SLINE2.
    MOVE SPACES TO SLINE3.
    MOVE SPACES TO SLINE4.
    MOVE SPACES TO SLINE5.
    MOVE SPACES TO SLINE6.
    MOVE SPACES TO SLINE7.
    MOVE SPACES TO SLINE8.
    MOVE SPACES TO SLINE9.


eject
1100-SCREEN-OF-EMPLOYERS.
    MOVE SCHEDUL-EMP-NAME (COUNTER) TO
         SF-EMPLOYER (COUNTER).
    MOVE SCHEDUL-POSITION-AVAIL (COUNTER) TO
         SF-POSITION (COUNTER).
    MOVE SCHEDUL-JOB-LOCATION (COUNTER) TO
         SF-LOCATION (COUNTER).
    MOVE SCHEDUL-SCHED-STATUS (COUNTER) TO
         SF-STATUS (COUNTER).
    MOVE SCHEDUL-BID-DATE-MONTH (COUNTER) TO
         TEMP-SF-BID-DATE-MONTH.
    MOVE SCHEDUL-BID-DATE-DAY (COUNTER) TO
         TEMP-SF-BID-DATE-DAY.

    IF (TEMP-SF-BID-DATE-MONTH) = LOW-VALUES THEN
       MOVE SPACES TO SF-BID-DATE (COUNTER)
    ELSE
       MOVE TEMP-SF-BID-DATE TO SF-BID-DATE (COUNTER).

    IF LOOP-NUM = 1 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE1
```

```
    ELSE
    IF LOOP-NUM = 2 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE2
    ELSE
    IF LOOP-NUM = 3 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE3
    ELSE
    IF LOOP-NUM = 4 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE4
    ELSE
    IF LOOP-NUM = 5 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE5
    ELSE
    IF LOOP-NUM = 6 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE6
    ELSE
    IF LOOP-NUM = 7 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE7
    ELSE
    IF LOOP-NUM = 8 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE8
    ELSE
    IF LOOP-NUM = 9 THEN
       MOVE SCRNOUT-FORMAT (COUNTER) TO SLINE9.

    COMPUTE LOOP-NUM = LOOP-NUM + 1.
    COMPUTE COUNTER = COUNTER + 1.

EXIT-SCREEN-OF-EMPLOYERS.
EJECT
2000-DISPLAY-SCREEN.
    MOVE 'EMPNO' TO SCRN-FIELD-NAME.
    MOVE ' '                        TO SENTRY.
*   CALL 'MPCUR' USING SCRN-NUMBER, SCRN-RETCODE, SCRN-FIELD-NAME.
    CALL 'SCRN1120' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.
    MOVE SPACES TO SYSMSG.
    MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.
EXIT-DISPLAY-SCREEN.
eject
3000-EVALUATE-PFKEYS.
    EVALUATE TRUE
        WHEN PF1
             CONTINUE
        WHEN PF3
             PERFORM 30300-PF3 THRU EXIT-PF3
             MOVE 'Y' TO PF3-PRESSED
        WHEN PF4
             PERFORM 30400-PF4 THRU EXIT-PF4
        WHEN PF7
             PERFORM 30700-PF7 THRU EXIT-PF7
        WHEN PF8
             PERFORM 30800-PF8 THRU EXIT-PF8
        WHEN OTHER
             MOVE INVALID-FUNCTION-KEY TO SYSMSG
             MOVE 'DISP-SCREEN' TO STEP-INDICATOR
    END-EVALUATE.
EXIT-EVALUATE-PFKEYS.
eject
30300-PF3.
    MOVE 'CICS    ' TO NEXT-PROGRAM-ID.
    MOVE 'XFER-CONTROL' TO STEP-INDICATOR.
EXIT-PF3.
30400-PF4.
    IF (SENTRY IS NOT NUMERIC) OR (SENTRY = ZERO) THEN
       MOVE INVALID-SELECTION TO SYSMSG
       MOVE SPACE TO SENTRY
```

```
           PERFORM 2000-DISPLAY-SCREEN
         ELSE
           MOVE SENTRY TO ENTRY-NUMBER
           COMPUTE ROWS-IN-TABLE-TO-COUNT-BACK = (10 - ENTRY-NUMBER)
           MOVE COUNTER TO COUNTER-BEFORE
           COMPUTE COUNTER = COUNTER - ROWS-IN-TABLE-TO-COUNT-BACK
           IF (SCHEDUL-EMP-NO (COUNTER) = '    ') THEN
             MOVE NO-EMPLOYER TO SYSMSG
             MOVE SPACE TO SENTRY
             MOVE COUNTER-BEFORE TO COUNTER
             PERFORM 2000-DISPLAY-SCREEN
           ELSE
             PERFORM 90200-PUT-DFHCOMMAREA
             MOVE 'XFER-CONTROL' TO STEP-INDICATOR.
       EXIT-PF4.

       30700-PF7.
           IF (COUNTER > 18)
             COMPUTE COUNTER = COUNTER - 18
             PERFORM 1000-PREPARE-SCREEN-OUTPUT
             PERFORM 2000-DISPLAY-SCREEN
           ELSE
             MOVE BEGINNING-PAGE TO SYSMSG
             PERFORM 2000-DISPLAY-SCREEN.
       EXIT-PF7.

       30800-PF8.
           IF SCHEDUL-EMP-NAME (COUNTER) = LOW-VALUES
             MOVE ENDING-PAGE TO SYSMSG
             PERFORM 2000-DISPLAY-SCREEN
           ELSE
           IF COUNTER < 493
             PERFORM 1000-PREPARE-SCREEN-OUTPUT
             PERFORM 2000-DISPLAY-SCREEN.
       EXIT-PF8.

       4000-TRANSFER-CONTROL.
      *    Populate those COMMAREA fields needed by next program.
           IF PF3-PRESSED = 'Y' THEN
             NEXT SENTENCE
           ELSE
           IF SCHEDUL-SCHED-STATUS(COUNTER) = 'BID    '
             MOVE 'CPPO0011' TO NEXT-PROGRAM-ID
           ELSE
             MOVE 'CPPO00XX' TO NEXT-PROGRAM-ID.
           MOVE 'CPPO0010' TO PREV-PROGRAM-ID.
           PERFORM 90200-PUT-DFHCOMMAREA.
           MOVE 0 TO RETURN-CODE.
       EXIT-TRANSFER-CONTROL.

       90100-GET-DFHCOMMAREA.
      *    MOVE '222222222' TO COMM-STUDENT-SSN.
           MOVE COMM-STUDENT-SSN TO LINK-SSN.
       EXIT-GET-DFHCOMMAREA-EXIT.

       90200-PUT-DFHCOMMAREA.
           MOVE SCHEDUL-EMP-NO (COUNTER) TO COMM-EMP-NUM.
           MOVE SCHEDUL-SCH-NO (COUNTER) TO COMM-SCH-NUM.
      *    MOVE SQL-STU-BID-POINTS-AVAIL TO COMM-STU-BID-PTS-AVAIL.
      *    MOVE SCHEDUL-INT-DATE1 (COUNTER) TO COMM-VIS-INT-DATE1.
      *    MOVE SCHEDUL-INT-DATE2 (COUNTER) TO COMM-VIS-INT-DATE2.
      *    MOVE SCHEDUL-INT-DATE3 (COUNTER) TO COMM-VIS-INT-DATE3.
      *    MOVE SCHEDUL-NO-SLOTS (COUNTER) TO COMM-VIS-TOT-SLOTS.
      *    MOVE SCHEDUL-NOTES (COUNTER)    TO COMM-VIS-NOTES.
      *    MOVE SCHEDUL-DEGREE-REQMT(COUNTER) TO COMM-VIS-DEGREE-REQMT.
      *    MOVE SCHEDUL-MAJOR-REQMT (COUNTER) TO COMM-VIS-MAJOR-REQMT.
```

```
      *    MOVE SCHEDUL-JOB-LOCATION(COUNTER) TO COMM-VIS-JOB-LOCATION.
      *    MOVE SCHEDUL-GRAD-DATE (COUNTER) TO COMM-VIS-GRAD-DATE.
      *    MOVE SCHEDUL-BID-DATE (COUNTER) TO COMM-VIS-BID-DATE.
           MOVE SQL-EMPLOYER-NAME          TO COMM-EMPLOYER-NAME.

      *    GOBACK.
      *EXIT-PUT-DFHCOMMAREA.

       95100-GET-SELECTED-STUDENT.
           EXEC SQL
                SELECT SSN,DEGREE,MAJOR1,MAJOR2,MAJOR3,
                       GRAD_MONTH,GRAD_YR,GPA_OVERALL,STATUS,
                       BID_POINTS_AVAIL
                INTO
                 :SQL-STU-SSN:NI           ,
                 :SQL-STU-DEGREE:NI        ,
                 :SQL-STU-MAJOR-1:NI       ,
                 :SQL-STU-MAJOR-2:NI       ,
                 :SQL-STU-MAJOR-3:NI       ,
                 :SQL-STU-GRAD-MONTH:NI    ,
                 :SQL-STU-GRAD-YR:NI       ,
                 :SQL-STU-GPA-OVERALL:NI   ,
                 :SQL-STU-SUSPEND-STATUS:NI,
                 :SQL-STU-BID-POINTS-AVAIL:NI
               FROM STUDENT
               WHERE SSN = :LINK-SSN
           END-EXEC.

           IF SQLCODE = 0 THEN
              MOVE 'OK' TO DB-STATUS-INDICATOR
           ELSE IF SQLCODE = 100
              MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
           ELSE GO TO 99200-DB-ABEND
           END-IF.
       EXIT-GET-SELECTED-STUDENT.
       EJECT
       95200-OPEN-SCHD-FOR-BROWSE.
           STRING SQL-STU-GRAD-MONTH SQL-STU-GRAD-YR DELIMITED BY SIZE
                  INTO SQL-STU-GRAD-DATE.

           EXEC SQL DECLARE C_SCHEDUL CURSOR FOR
             SELECT EMP_NO,SCH_NO,
      *             DATE_1_INTERVIEW,DATE_2_INTERVIEW,
      *             DATE_3_INTERVIEW,
                    DEGREE_REQMT,MAJORS,STATUS,
                    POSITION,LOCATION,
      *             GRAD_WHEN,
                    BID_BY_DATE
      *             NUM_TOTAL_SLOTS,ADMIN_NOTES
               FROM VISIT
      *        WHERE (MAJORS LIKE '%:SQL-STU-MAJOR-1%'
      *        WHERE (MAJORS LIKE %:SQL-STU-MAJOR-1%
               WHERE (MAJORS = :SQL-STU-MAJOR-1
                OR MAJORS = :SQL-STU-MAJOR-2
                OR MAJORS = :SQL-STU-MAJOR-3)
                 AND DEGREE_REQMT =
                :SQL-STU-DEGREE AND BID_BY_DATE >= :HOST-BID-DATE AND
                (STATUS = 'BID    ' or STATUS = 'OPEN   ') AND
                GRAD_WHEN = :SQL-STU-GRAD-DATE AND EMP_NO NOT IN
                    (SELECT EMP_NO FROM BID
                     WHERE SSN = :SQL-STU-SSN AND STATUS = 'CLOSED')
           END-EXEC.
           EXEC SQL OPEN C_SCHEDUL END-EXEC.

           IF SQLCODE = 0 THEN
               MOVE 'OK' TO DB-STATUS-INDICATOR
```

```
    ELSE
        GO TO 99200-DB-ABEND.
EXIT-OPEN-SCHD-FOR-BROWSE.
EJECT
95300-GET-NEXT-SCHEDULE.
    EXEC SQL FETCH C_SCHEDUL INTO
         :SQL-SCHEDUL-EMP-NO:NI,
         :SQL-SCHEDUL-SCH-NO:NI,
*        :SQL-SCHEDUL-INT-DATE1:NI,
*        :SQL-SCHEDUL-INT-DATE2:NI,
*        :SQL-SCHEDUL-INT-DATE3:NI,
         :SQL-SCHEDUL-DEGREE-REQMT:NI,
         :SQL-SCHEDUL-MAJOR-REQMT:NI,
         :SQL-SCHEDUL-SCHED-STATUS:NI,
         :SQL-SCHEDUL-POSITION-AVAIL:NI,
         :SQL-SCHEDUL-JOB-LOCATION:NI,
*        :SQL-SCHEDUL-GRAD-DATE:NI,
         :SQL-SCHEDUL-BID-DATE:NI
*        :SQL-SCHEDUL-NO-SLOTS:NI,
*        :SQL-SCHEDUL-NOTES:NI
    END-EXEC.
    IF SQLCODE  = 0 THEN
    MOVE 'OK' TO DB-STATUS-INDICATOR
    MOVE SQL-SCHEDUL-EMP-NO      TO SCHEDUL-EMP-NO (EMP-IDX)
    MOVE SQL-SCHEDUL-SCH-NO      TO SCHEDUL-SCH-NO (EMP-IDX)
*   MOVE SQL-SCHEDUL-INT-DATE1  TO SCHEDUL-INT-DATE1(EMP-IDX)
*   MOVE SQL-SCHEDUL-INT-DATE2  TO SCHEDUL-INT-DATE2(EMP-IDX)
*   MOVE SQL-SCHEDUL-INT-DATE3  TO SCHEDUL-INT-DATE3(EMP-IDX)
    MOVE SQL-SCHEDUL-DEGREE-REQMT
                                TO SCHEDUL-DEGREE-REQMT (EMP-IDX)
    MOVE SQL-SCHEDUL-MAJOR-REQMT TO SCHEDUL-MAJOR-REQMT (EMP-IDX)
    MOVE SQL-SCHEDUL-SCHED-STATUS
                                TO SCHEDUL-SCHED-STATUS (EMP-IDX)
    MOVE SQL-SCHEDUL-POSITION-AVAIL
                                TO SCHEDUL-POSITION-AVAIL (EMP-IDX)
    MOVE SQL-SCHEDUL-JOB-LOCATION
                                TO SCHEDUL-JOB-LOCATION (EMP-IDX)
*   MOVE SQL-SCHEDUL-GRAD-DATE TO SCHEDUL-GRAD-DATE (EMP-IDX)
    MOVE SQL-SCHEDUL-BID-DATE  TO TEMP-BID-DATE
    MOVE TEMP-BID-DAY TO SCHEDUL-BID-DATE-DAY (EMP-IDX)
    MOVE TEMP-BID-MONTH TO SCHEDUL-BID-DATE-MONTH (EMP-IDX)
*   MOVE SQL-SCHEDUL-NO-SLOTS TO SCHEDUL-NO-SLOTS (EMP-IDX)
*   MOVE SQL-SCHEDUL-NOTES     TO SCHEDUL-NOTES (EMP-IDX)
    PERFORM 95400-GET-SELECTED-EMPLOYER
    ELSE IF SQLCODE = 100
        MOVE 'EOF' TO DB-STATUS-INDICATOR
    ELSE
        GO TO 99200-DB-ABEND.
EXIT-GET-NEXT-SCHEDULE.
EJECT
95400-GET-SELECTED-EMPLOYER.
    MOVE SCHEDUL-EMP-NO (EMP-IDX) TO HOLD-EMP-NO.

    EXEC SQL
        SELECT NAME INTO
         :SQL-EMPLOYER-NAME:NI
        FROM EMPLOYER
        WHERE EMP_NO = :HOLD-EMP-NO
    END-EXEC.

    IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR
        MOVE SQL-EMPLOYER-NAME TO SCHEDUL-EMP-NAME (EMP-IDX)
    ELSE IF SQLCODE = 100
        MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
    ELSE
```

```
        GO TO 99200-DB-ABEND.
EXIT-GET-SELECTED-EMPLOYER.
95500-CLOSE-SCHEDUL-CURSOR.
    EXEC SQL CLOSE C_SCHEDUL END-EXEC.
    IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR.
EXIT-CLOSE-SCHEDUL-CURSOR.
EJECT
99000-ABNORMAL-TERMINATION.
*   All abnormal terminations handled from here.
                                                                  0
99200-DB-ABEND.
*****************************************************************  00623000
* THE FOLLOWING ROUTINE PRINTS THE SQLCA STRUCTURE:             *  00624000
*                                                               *  00625000
*  - SQLCODE = SQL RETURN CODE                                  *  00626000
*  - SQLERRM = SQL ERROR MESSAGE                                *  00627000
*  - SQLERRP = MODULE DETECTING ERROR                           *  00628000
*  - SQLERRD = INTERNAL ERROR VALUES                            *  00629000
*  - SQLWARN = SQL WARNING STRUCTURE                            *  00630000
*                                                               *  00631000
*****************************************************************  00632000
      DISPLAY '*********************************' UPON CONSOLE.   00633000
      DISPLAY '* PROGRAM ERROR ROUTINE ENTERED *' UPON CONSOLE.   00634000
      DISPLAY '* CHECK SYSPRINT FOR ERROR CODES*' UPON CONSOLE.   00635000
      DISPLAY '* CHANGES WILL BE BACKED OUT    *' UPON CONSOLE.   00636000
      DISPLAY '*********************************' UPON CONSOLE.   00637000
      MOVE SQLCODE TO DECODED-SQLCODE.                            00638000
      DISPLAY 'PROGRAM ERROR ROUTINE ENTERED'.                    00639000
      DISPLAY '        *******'.                                  00640000
      DISPLAY 'A PROBLEM HAS BEEN DETECTED IN THE '.              00641000
      DISPLAY STEP-INDICATOR, ' PARAGRAPH.'.                      00642000
      DISPLAY 'THE FOLLOWING ERROR CODES SHOULD AID YOU IN'.      00643000
      DISPLAY 'PROBLEM DETERMINATION OF THE SQL STATEMENT.'.      00644000
      DISPLAY '        *******'.                                  00645000
      DISPLAY 'SQLCODE : ' DECODED-SQLCODE.                       00646000
      DISPLAY 'SQLERRM : ' SQLERRMC.                              00647000
      DISPLAY 'SQLERRP : ' SQLERRP.                               00648000
      PERFORM ERRD VARYING INDX2 FROM 1 BY 1 UNTIL INDX2 = 7.     00649000
      IF SQLWARN0 NOT EQUAL 'W'                                   00650000
          THEN GO TO 99000-BACKOUT,                               00651000
          ELSE DISPLAY 'SQLWARN0: ' SQLWARN0,                     00652000
               DISPLAY 'SQLWARN1: ' SQLWARN1,                     00653000
               DISPLAY 'SQLWARN2: ' SQLWARN2,                     00654000
               DISPLAY 'SQLWARN3: ' SQLWARN3,                     00655000
               DISPLAY 'SQLWARN4: ' SQLWARN4,                     00656000
               DISPLAY 'SQLWARN5: ' SQLWARN5,                     00657000
               DISPLAY 'SQLWARN6: ' SQLWARN6,                     00658000
               DISPLAY 'SQLWARN7: ' SQLWARN7,                     00659000
               DISPLAY 'SQLWARN8: ' SQLWARN8,                     00660000
               DISPLAY 'SQLWARN9: ' SQLWARN9,                     00661000
               DISPLAY 'SQLWARNA: ' SQLWARNA,                     00662000
               GO TO 99000-BACKOUT.                               00663000

ERRD.
      MOVE SQLERRD (INDX2) TO DECODED-SQLERRD (INDX2).             0
      DISPLAY 'SQLERRD', INDX2, ': ', DECODED-SQLERRD (INDX2).    00666000
                                                                   0621000
EXIT-ERROR-ABEND.
eject
99000-BACKOUT.                                                     00667000
*****************************************************************  00668000
* 'WHENEVER' RESET TO 'CONTINUE' IN THE EVENT THAT THE ROLLBACK *  00669000
* WORK STATEMENT FAILS TO AVOID LOOP IN ERROR ROUTINE.          *  00670000
*****************************************************************  00671000
                                                                   00672000
    MOVE 'ABEND - BACKING OUT' TO STEP-INDICATOR.                   0674000
                                                                   00675000
```

```
EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.                   00676000
EXEC SQL ROLLBACK WORK END-EXEC.                                00677000
STOP RUN.
```

3) Source Code of Module: CPPO0011

```
*****************************************************************
*  CPPO0011: Place bid for selected employer.
*****************************************************************

 IDENTIFICATION DIVISION.
 PROGRAM-ID.  CPPO0011.
 AUTHOR.      Cory Snavely CS38SANU.

*MODIFIED 10/92 S.PETER. EMBEDDED STATEMENTS
*MODIFIED 12/16/92 S.Peter. 2 if stmts in proc 1000-.
                            fixed STATUS in 30120-.
                            fixed STATUS in 30130-.
*MODIFIED 12/20/92 S.Peter. changed SQL-BID-STATUS from 6 to 8 char
*MODIFIED
*MODIFIED
* PROGRAM FUNCTIONAL REQUIREMENTS
* Insert a listing of the program's functional requirements here.
*
*  Accept an employer number, schedule number, and student ID
*  number from the calling program.  Access the CPPO schedule
*  and display the corresponding schedule information.
*
*  Accept an amount of bid points to be submitted.  Add the
*  bid or change an existing bid to the amount specified.
*  Delete an existing bid if 0 is entered.
*
*  Update the student's bid points remaining.
*
*  Assumption:
*
*    Any bids submitted will be valid; invalidity of bids
*    will be checked by the calling program.

 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
     C01 IS TOP-OF-PAGE.

 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

 DATA DIVISION.
 FILE SECTION.

 WORKING-STORAGE SECTION.                                          MP00020

     EXEC SQL BEGIN DECLARE SECTION END-EXEC.

 01  DFH-EMP-NUM                      PIC X(5).
 01  DFH-SCH-NUM                      PIC X(2).
 01  DFH-STUD-ID                      PIC X(9).
 01  SQL-STU-BID-POINTS-AVAIL         PIC S9(04) COMP.

*SQL-BID ENTITY

 01  SQL-BID-SSN                      PIC X(09).
 01  SQL-BID-EMP-NO                   PIC X(05).
 01  SQL-BID-SCH-NO                   PIC X(2).
 77  SQL-BID-POINTS-BID               PIC S9(4) COMP.
 01  SQL-BID-STATUS                   PIC X(08).
 01  SQL-BID-DATE-OF-BID              PIC X(10).
 01  SQL-BID-TIME-OF-BID              PIC X(08).

*SQL-EMPLOYER ENTITY                                               MP00030

 01  SQL-EMP-NAME                     PIC X(40).                  MP00080
```

```
* SQL-SCHEDUL ENTITY
 01  SQL-SCHEDUL-EMP-NO               PIC X(5).
 01  SQL-SCHEDUL-SCH-NO               PIC X(2).
 01  SQL-SCHEDUL-INT-DATE1            PIC X(10).
 01  SQL-SCHEDUL-INT-DATE2            PIC X(10).
 01  SQL-SCHEDUL-INT-DATE3            PIC X(10).
 01  SQL-SCHEDUL-TOT-SLOTS            PIC S9(4) COMP.
 01  SQL-SCHEDUL-DEGREE-REQMT         PIC X(3).
 01  SQL-SCHEDUL-MAJOR-REQMT          PIC X(48).
*01  SQL-SCHEDUL-MIN-GPA              PIC S9V9 COMP-3.
 01  SQL-SCHEDUL-JOB-LOCATION         PIC X(34).
 01  SQL-SCHEDUL-GRAD-DATE            PIC XXX.
 01  SQL-SCHEDUL-BID-DATE             PIC X(10).
 01  SQL-SCHEDUL-NOTES                PIC X(254).
 01  SQL-SCHEDUL-ADDITIONAL-REQMTS    PIC X(60).

*    STUDENT COLUMNS FROM STUDENT TABLE.                          TU00020
*    IDENTIFICATION DATA                                          TU00030
*    CPPO DATA                                                    TU00480
 77  SQL-STU-BID-POINTS               PIC S9(4) COMP.             TU00490

 77  NI                               PIC S9(4) COMP.
                                                                  TU01760
     EXEC SQL END DECLARE SECTION  END-EXEC.

*    NUMERIC BID POINTS FIELD FOR COMPARISON
 01  BID-POINTS-BID                   PIC 9(4).

*    NUMERIC BID POINTS FIELD TO HOLD AMOUNT OF OLD BID
 01  OLD-POINTS-BID                   PIC 9(4).

*    GRADUATION DATE FIELD FOR STRING MANIPULATION
 01  SCHEDUL-GRAD-DATE.
     05  G-MONTH                      PIC X.
     05  G-YEAR                       PIC XX.

*    MAJOR REQUIREMENT FIELD FOR STRING MANIPULATION
 01  SCHEDUL-MAJOR-REQMT.
     05 S-MAJOR-1                     PIC X(20).
     05 S-MAJOR-2                     PIC X(20).
     05 S-MAJOR-3                     PIC X(10).
                                                                  TU01750
     EXEC SQL INCLUDE SQLCA END-EXEC.                             0100000

*    Additional variables for abnormal termination.
 01  DECODED-SQLCODE          PIC --------999.                    0107000
                                                                  0108000
 01  ARRAY-SQLERRD.                                              0109000
     02 DECODED-SQLERRD      PIC --------999 OCCURS 6 TIMES.     0110000
 01  INDX2                    PIC S9(1) SYNC USAGE IS COMP.       0111000
 01  INDXPIC                  PIC ZZZ9.                           0112000

*****************************************************************  0102000
*   PROGRAM VARIABLE DECLARATION SECTION                          *  0104000
*****************************************************************

 01  STEP-INDICATOR           PIC X(30) VALUE 'PREP-SCREEN'.      001050
 01  DB-STATUS-INDICATOR      PIC X(8)  VALUE SPACES.
 01  EDIT-STATUS-INDICATOR    PIC X(8)  VALUE SPACES.
 01  THIS-PROGRAM-ID          PIC X(8)  VALUE 'CPPO0011'.
 01  SCRN-NAME                PIC X(8)  VALUE 'SCRN1121'.
 01  SCRN-RETCODE             PIC S9(7) COMP.
 01  SCRN-FIELD-NAME          PIC X(7).
 01  FIELDNAME                PIC X(7).
 01  BID-SUBMIT-STATUS        PIC X(9)  VALUE SPACES.
```

```
01  SQL-SCRATCH-DATE.
    05  SQL-CENTURY          PIC XX.
    05  SQL-YEAR             PIC XX.
    05  FILLER               PIC X.
    05  SQL-MONTH            PIC XX.
    05  FILLER               PIC X.
    05  SQL-DAY              PIC XX.
01  SYSTEM-DATE.
    05  S-YEAR               PIC XX.
    05  S-MONTH              PIC XX.
    05  S-DAY                PIC XX.
01  SYSTEM-TIME.
    05  S-HOURS              PIC XX.
    05  S-MINUTES            PIC XX.
    05  S-SECONDS            PIC X(2).
    05  S-HUNDRTH-SEC        PIC X(2).

 01 SCRN-NUMBER    PIC     9(7) COMP.                                    UN00020
 01 SCRN-FIELDS.                                                        JUN00010
    05 PGMID          PIC     X(8).                                     JUN00030
    05 SCRNID         PIC     X(8).                                     JUN00040
    05 SCRDATE        PIC     X(8).                                     JUN00050
    05 SCRTIME        PIC     X(5).                                     JUN00060
    05 EMPNUM         PIC     X(5).                                     JUN00070
    05 SCHNUM         PIC     X(2).                                     JUN00080
    05 EMP            PIC    X(30).                                     JUN00090
    05 LOC            PIC    X(25).                                     JUN00100
    05 MAJ1           PIC    X(20).                                     JUN00110
    05 DEGREE         PIC     X(5).                                     JUN00120
    05 MAJ2           PIC    X(20).                                     JUN00130
    05 GRADATE        PIC     X(5).                                     JUN00140
    05 MAJ3           PIC    X(20).                                      UN00150
    05 INTDAT1        PIC    X(10).                                      UN00150
    05 INTDAT2        PIC    X(10).                                      UN00150
    05 INTDAT3        PIC    X(10).                                     JUN00160
    05 BIDDATE        PIC    X(10).                                     JUN00170
    05 NOSLOTS        PIC      ZZ9.                                     JUN00180
*   05 PREFGPA        PIC     9.99.                                     JUN00190
    05 MISCNOT        PIC    X(60).                                     JUN00200
    05 BIDPLFT        PIC     ZZZ9.                                     JUN00210
    05 BIDPNT         PIC     X(4).
*NOTE: BIDPNT IS CHAR SO THAT UNDERSCORES CAN BE PUSHED
*      TO SCREEN.  THIS LETS USER KNOW WHERE ENTRY AREAS ARE &
*      KEEPS THE CONSISTENCY OF SCREENS. MODIFIED S.PETER 10/11/92
    05 SYSMSG         PIC    X(70).                                     JUN00220
 01 SCRN-KEY       PIC     X(8).                                        UN00230
    88  PF1                          VALUE 'PF01'.
    88  PF2                          VALUE 'PF02'.
    88  PF3                          VALUE 'PF03'.
    88  PF4                          VALUE 'PF04'.
    88  PF5                          VALUE 'PF05'.
    88  PF6                          VALUE 'PF06'.
    88  PF7                          VALUE 'PF07'.
    88  PF8                          VALUE 'PF08'.
    88  PF9                          VALUE 'PF09'.
    88  PF10                         VALUE 'PF10'.
    88  PF11                         VALUE 'PF11'.
    88  PF12                         VALUE 'PF12'.
    88  RETURN-KEY                   VALUE 'ENTER'.

 LINKAGE SECTION.

 COPY DFHCOMM.

 PROCEDURE DIVISION USING DFHCOMMAREA.
     EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
```

```
 0000-MAINLINE SECTION.
*    PERFORM 00001-SIMULATE-DFH.
     PERFORM 90100-GET-DFHCOMMAREA.
     CALL 'MLOAD' USING SCRN-NUMBER SCRN-RETCODE SCRN-NAME.
     PERFORM UNTIL STEP-INDICATOR = 'XFER-CONTROL'
         EVALUATE TRUE
             WHEN STEP-INDICATOR = 'PREP-SCREEN'
                  PERFORM 1000-PREPARE-SCREEN-OUTPUT
             WHEN STEP-INDICATOR = 'DISP-SCREEN'
                  PERFORM 2000-DISPLAY-SCREEN
             WHEN STEP-INDICATOR = 'EVAL-PFKEYS'
                  PERFORM 3000-EVALUATE-PFKEYS
             WHEN OTHER
                  DISPLAY 'STEP INDICATOR NOT SET'
                  PERFORM 99000-BACKOUT
         END-EVALUATE
     END-PERFORM.
     PERFORM 4000-TRANSFER-CONTROL.
     CALL 'MPURGE' USING SCRN-NUMBER SCRN-RETCODE.
     GOBACK.
 EXIT-MAINLINE.

 1000-PREPARE-SCREEN-OUTPUT SECTION.

     MOVE THIS-PROGRAM-ID TO PGMID.

     MOVE SCRN-NAME TO SCRNID.

     ACCEPT SYSTEM-DATE FROM DATE
     STRING S-MONTH '/' S-DAY '/' S-YEAR DELIMITED BY SIZE
       INTO SCRDATE.

     ACCEPT SYSTEM-TIME FROM TIME
     STRING S-HOURS ':' S-MINUTES DELIMITED BY SIZE INTO
       SCRTIME.

     MOVE '____' TO BIDPNT.

     EXEC SQL
         SELECT EMP_NO, SCH_NO, LOCATION, DEGREE_REQMT,
                GRAD_WHEN,DATE_1_INTERVIEW,DATE_2_INTERVIEW,
                DATE_3_INTERVIEW,BID_BY_DATE,
                ADDITIONAL_REQMTS, NUM_TOTAL_SLOTS, MAJORS
                INTO
              :SQL-SCHEDUL-EMP-NO:NI,
              :SQL-SCHEDUL-SCH-NO:NI,
              :SQL-SCHEDUL-JOB-LOCATION:NI,
              :SQL-SCHEDUL-DEGREE-REQMT:NI,
              :SQL-SCHEDUL-GRAD-DATE:NI,
              :SQL-SCHEDUL-INT-DATE1:NI,
              :SQL-SCHEDUL-INT-DATE2:NI,
              :SQL-SCHEDUL-INT-DATE3:NI,
              :SQL-SCHEDUL-BID-DATE:NI,
**            :SQL-SCHEDUL-MIN-GPA:NI,
              :SQL-SCHEDUL-ADDITIONAL-REQMTS:NI,
              :SQL-SCHEDUL-TOT-SLOTS:NI,
              :SQL-SCHEDUL-MAJOR-REQMT:NI
         FROM VISIT
         WHERE EMP_NO = :DFH-EMP-NUM
           AND SCH_NO = :DFH-SCH-NUM
     END-EXEC.

     IF SQLCODE NOT = 0
       THEN
         GO TO 99200-DB-ABEND.
```

```
*    STRING DFH-EMP-NUM '-' DFH-SCH-NUM
*      DELIMITED BY SIZE
*      INTO EMSCHNO
*    END-STRING.
     MOVE SQL-SCHEDUL-EMP-NO TO EMPNUM.
     MOVE SQL-SCHEDUL-SCH-NO TO SCHNUM.
     MOVE SQL-SCHEDUL-JOB-LOCATION TO LOC.
     MOVE SQL-SCHEDUL-DEGREE-REQMT TO DEGREE.
     MOVE SQL-SCHEDUL-GRAD-DATE TO SCHEDUL-GRAD-DATE.
     STRING G-MONTH '/' G-YEAR DELIMITED BY SIZE INTO GRADATE.
     MOVE SQL-SCHEDUL-MAJOR-REQMT TO SCHEDUL-MAJOR-REQMT.
     MOVE S-MAJOR-1 TO MAJ1.
     MOVE S-MAJOR-2 TO MAJ2.
     MOVE S-MAJOR-3 TO MAJ3.
     MOVE SQL-SCHEDUL-INT-DATE1 TO SQL-SCRATCH-DATE.
     STRING SQL-MONTH '/' SQL-DAY '/' SQL-YEAR
              DELIMITED BY SIZE INTO INTDAT1.
*modified the following two if stmts to prevent '/'s going
*to screen.   Steve Peter 12/16/92.
     MOVE SQL-SCHEDUL-INT-DATE2 TO SQL-SCRATCH-DATE.
     IF (SQL-SCRATCH-DATE  = '          ') OR
        (SQL-SCRATCH-DATE  = LOW-VALUES)
              MOVE '          ' TO INTDAT2
      ELSE
              STRING SQL-MONTH '/' SQL-DAY '/' SQL-YEAR
              DELIMITED BY SIZE INTO INTDAT2.
     MOVE SQL-SCHEDUL-INT-DATE3 TO SQL-SCRATCH-DATE.
     IF (SQL-SCRATCH-DATE  = '          ') OR
        (SQL-SCRATCH-DATE  = LOW-VALUES)
              MOVE '          ' TO INTDAT3
      ELSE
     STRING SQL-MONTH '/' SQL-DAY '/' SQL-YEAR
              DELIMITED BY SIZE INTO INTDAT3.
     MOVE SQL-SCHEDUL-BID-DATE TO SQL-SCRATCH-DATE.
     STRING SQL-MONTH '/' SQL-DAY '/' SQL-YEAR
              DELIMITED BY SIZE INTO BIDDATE.
*    MOVE SQL-SCHEDUL-MIN-GPA TO PREFGPA.
     MOVE SQL-SCHEDUL-TOT-SLOTS TO NOSLOTS.
     MOVE SQL-SCHEDUL-ADDITIONAL-REQMTS TO MISCNOT.

     EXEC SQL                                                    MP00020
          SELECT NAME INTO                                       MP00030
               :SQL-EMP-NAME:NI                                  MP00080
          FROM EMPLOYER                                          MP00120
          WHERE EMP_NO = :DFH-EMP-NUM
     END-EXEC.                                                   MP00140
                                                                 MP00150
     IF SQLCODE NOT = 0
       THEN
         GO TO 99200-DB-ABEND.

     MOVE SQL-EMP-NAME          TO  EMP.                         MP00210

     EXEC SQL
          SELECT BID_POINTS_AVAIL INTO
               :SQL-STU-BID-POINTS-AVAIL:NI
          FROM STUDENT
          WHERE SSN = :DFH-STUD-ID
     END-EXEC.

     IF SQLCODE NOT = 0
       THEN
         GO TO 99200-DB-ABEND.

     MOVE SQL-STU-BID-POINTS-AVAIL TO BIDPLFT.
```

```
     MOVE ALL SPACES TO SYSMSG.

     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.

EXIT-PREPARE-SCREEN-OUTPUT.

2000-DISPLAY-SCREEN SECTION.

     CALL 'SCRN1121' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.

     MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.
EXIT-DISPLAY-SCREEN.

3000-EVALUATE-PFKEYS SECTION.
     EVALUATE TRUE
          WHEN RETURN-KEY
               MOVE ALL SPACES TO SYSMSG
               MOVE 'DISP-SCREEN' TO STEP-INDICATOR
          WHEN PF1
               MOVE 'DISP-SCREEN' TO STEP-INDICATOR
          WHEN PF3
               MOVE PREV-PROGRAM-ID TO NEXT-PROGRAM-ID
               MOVE 'XFER-CONTROL' TO STEP-INDICATOR
          WHEN PF4
               PERFORM 30100-PF4
          WHEN OTHER
               MOVE SPACES TO SYSMSG
               MOVE 'You pressed an invalid function key.'
                 TO SYSMSG
               MOVE 'DISP-SCREEN' TO STEP-INDICATOR
     END-EVALUATE.
EXIT-EVALUATE-PFKEYS.

4000-TRANSFER-CONTROL SECTION.
*    Populate those COMMAREA fields needed by next program.
     MOVE PREV-PROGRAM-ID TO NEXT-PROGRAM-ID.
     MOVE 'CPP00011' TO PREV-PROGRAM-ID.
     PERFORM 90200-PUT-DFHCOMMAREA.
     MOVE 0 TO RETURN-CODE.
EXIT-TRANSFER-CONTROL.

30100-PF4 SECTION.

     MOVE SQL-STU-BID-POINTS-AVAIL TO SQL-STU-BID-POINTS.
     MOVE SPACES TO SYSMSG
*    IF BIDPNT < ZERO
*      THEN
*        MOVE 'You cannot submit a negative bid.' TO SYSMSG
*        MOVE ZEROES TO BIDPNT
     MOVE BIDPNI TO BID-POINTS-BID.
     IF BID-POINTS-BID IS NOT NUMERIC
       THEN
         MOVE 'Your entry was not a number.' TO SYSMSG
         MOVE '____' TO BIDPNT
       ELSE
*        MOVE BIDPNT TO BID-POINTS-BID
         PERFORM 30110-CHECK-PREVIOUS-BID
         EVALUATE BID-SUBMIT-STATUS
           WHEN 'DUPLICATE'
             STRING 'The above bid has already been'
               ' submitted.' DELIMITED BY SIZE INTO SYSMSG
           WHEN 'NEW'
             IF BID-POINTS-BID = ZERO
               THEN
                 STRING 'You have not submitted a bid for this'
```

```
                    ' employer and schedule.' DELIMITED BY SIZE
                    INTO SYSMSG
                ELSE
                  IF SQL-STU-BID-POINTS < BID-POINTS-BID
                    THEN
                      STRING 'You do not have enough bid points.'
                          DELIMITED BY SIZE INTO SYSMSG
                    ELSE
                      PERFORM 30120-ADD-BID
                      PERFORM 30140-UPDATE-STU-RECORD
                      MOVE SQL-STU-BID-POINTS TO BIDPLFT
                      STRING 'The above bid has been'
                          ' successfully submitted.' DELIMITED
                          BY SIZE INTO SYSMSG
                  END-IF
              END-IF
            WHEN 'UPDATE'
              IF BID-POINTS-BID = ZERO
                THEN
                  PERFORM 30150-DELETE-BID
                  PERFORM 30140-UPDATE-STU-RECORD
                  MOVE SQL-STU-BID-POINTS TO BIDPLFT
                  STRING 'Your bid for this employer and schedule'
                    ' has been successfully deleted.' DELIMITED BY
                    SIZE INTO SYSMSG
                ELSE
                  IF (SQL-STU-BID-POINTS + OLD-POINTS-BID) <
                      BID-POINTS-BID
                    THEN
                      STRING 'You do not have enough bid-'
                          ' points.' DELIMITED BY SIZE INTO
                          SYSMSG
                    ELSE
                      PERFORM 30130-UPDATE-BID
                      PERFORM 30140-UPDATE-STU-RECORD
                      MOVE SQL-STU-BID-POINTS TO BIDPLFT
                      STRING 'Your bid for this employer and'
                          ' schedule has been successfully'
                          ' changed.' DELIMITED BY SIZE INTO
                          SYSMSG
                  END-IF
              END-IF
          END-EVALUATE
      END-IF

      MOVE 'DISP-SCREEN' TO STEP-INDICATOR.

  EXIT-30100-PF4.

  30110-CHECK-PREVIOUS-BID SECTION.

      MOVE ZEROES TO SQL-BID-POINTS-BID.

      EXEC SQL                                                    MP00020
          SELECT POINTS_BID INTO                                  MP00030
              :SQL-BID-POINTS-BID:NI                              MP00080
          FROM BID                                                MP00120
          WHERE EMP_NO = :DFH-EMP-NUM
            AND SCH_NO = :DFH-SCH-NUM
            AND SSN    = :DFH-STUD-ID
      END-EXEC.                                                   MP00140

      IF SQLCODE NOT = 0 AND SQLCODE NOT = 100
        THEN
          GO TO 99200-DB-ABEND.
```

```
      MOVE SQL-BID-POINTS-BID TO OLD-POINTS-BID.

      IF SQLCODE = 100
        THEN
          MOVE 'NEW' TO BID-SUBMIT-STATUS
        ELSE
          IF OLD-POINTS-BID = BID-POINTS-BID AND
             BID-POINTS-BID NOT = ZERO
            THEN
              MOVE 'DUPLICATE' TO BID-SUBMIT-STATUS
            ELSE
              MOVE 'UPDATE' TO BID-SUBMIT-STATUS
          END-IF
      END-IF.
  EXIT-30110-CHECK-PREVIOUS-BID.

  30120-ADD-BID SECTION.

      MOVE DFH-STUD-ID    TO  SQL-BID-SSN
      MOVE DFH-EMP-NUM    TO  SQL-BID-EMP-NO
      MOVE DFH-SCH-NUM    TO  SQL-BID-SCH-NO
      MOVE BID-POINTS-BID  TO  SQL-BID-POINTS-BID
      MOVE 'UNPROCED'      TO  SQL-BID-STATUS
      STRING '19' S-YEAR '-' S-MONTH '-' S-DAY DELIMITED BY SIZE
        INTO SQL-BID-DATE-OF-BID.
      STRING S-HOURS '.' S-MINUTES '.' S-SECONDS DELIMITED BY SIZE
        INTO SQL-BID-TIME-OF-BID.

      EXEC SQL
          INSERT INTO BID (SSN, EMP_NO, SCH_NO, POINTS_BID,
                 STATUS,TIME_OF_BID, DATE_OF_BID)
          VALUES
           (:SQL-BID-SSN,
            :SQL-BID-EMP-NO,
            :SQL-BID-SCH-NO,
            :SQL-BID-POINTS-BID,
            :SQL-BID-STATUS,
            :SQL-BID-TIME-OF-BID,
            :SQL-BID-DATE-OF-BID)
      END-EXEC.

      IF SQLCODE NOT = 0
        THEN
          GO TO 99200-DB-ABEND.

  EXIT-30120-ADD-BID.

  30130-UPDATE-BID SECTION.

      MOVE BID-POINTS-BID   TO  SQL-BID-POINTS-BID
      STRING '19' S-YEAR '-' S-MONTH '-' S-DAY DELIMITED BY SIZE
        INTO SQL-BID-DATE-OF-BID.
      STRING S-HOURS '.' S-MINUTES '.' S-SECONDS DELIMITED BY SIZE
        INTO SQL-BID-TIME-OF-BID.

      EXEC SQL
          UPDATE BID
            SET POINTS_BID = :SQL-BID-POINTS-BID,
                DATE_OF_BID = :SQL-BID-DATE-OF-BID,
                TIME_OF_BID = :SQL-BID-TIME-OF-BID,
                STATUS      = 'UNPROCED'
            WHERE EMP_NO = :DFH-EMP-NUM
              AND SCH_NO = :DFH-SCH-NUM
              AND SSN    = :DFH-STUD-ID
      END-EXEC.
```

```
           IF SQLCODE NOT = 0
             THEN
               GO TO 99200-DB-ABEND.

       EXIT-30130-UPDATE-BID.

       30140-UPDATE-STU-RECORD SECTION.

           ADD OLD-POINTS-BID TO SQL-STU-BID-POINTS.

           SUBTRACT BID-POINTS-BID FROM SQL-STU-BID-POINTS.

           EXEC SQL                                                    TU00040
               UPDATE STUDENT
                 SET BID_POINTS_AVAIL = :SQL-STU-BID-POINTS            TU00440
               WHERE SSN = :DFH-STUD-ID
           END-EXEC.                                                   TU00790
                                                                       TU00800
           IF SQLCODE NOT = 0
             THEN
               GO TO 99200-DB-ABEND.

       EXIT-30140-UPDATE-STU-RECORD.

       30150-DELETE-BID SECTION.

           EXEC SQL                                                    TU00040
               DELETE FROM BID
               WHERE SSN = :DFH-STUD-ID
                 AND EMP_NO = :DFH-EMP-NUM
                 AND SCH_NO = :DFH-SCH-NUM
           END-EXEC.                                                   TU00790
                                                                       TU00800
           IF SQLCODE NOT = 0
             THEN
               GO TO 99200-DB-ABEND.

       EXIT-30140-UPDATE-STU-RECORD.

       90100-GET-DFHCOMMAREA SECTION.
      *    MOVE '222222222' TO COMM-STUDENT-SSN.
      *    MOVE '  999' TO COMM-EMP-NUM.
      *    MOVE ' 1' TO COMM-SCH-NUM.
           MOVE COMM-STUDENT-SSN TO DFH-STUD-ID.
           MOVE COMM-EMP-NUM TO DFH-EMP-NUM.
           MOVE COMM-SCH-NUM TO DFH-SCH-NUM.
      *    MOVE COMM-STU-BID-PTS-AVAIL TO SQL-STU-BID-POINTS-AVAIL.
      *    MOVE COMM-EMPLOYER-NAME    TO SQL-EMP-NAME.

      *    MOVE COMM-VIS-INT-DATE1    TO SQL-SCHEDUL-INT-DATE1.
      *    MOVE COMM-VIS-INT-DATE2    TO SQL-SCHEDUL-INT-DATE2.
      *    MOVE COMM-VIS-INT-DATE3    TO SQL-SCHEDUL-INT-DATE3.
      *    MOVE COMM-VIS-TOT-SLOTS    TO SQL-SCHEDUL-TOT-SLOTS.
      *    MOVE COMM-VIS-DEGREE-REQMT TO SQL-SCHEDUL-DEGREE-REQMT.
      *    MOVE COMM-VIS-MAJOR-REQMT  TO SQL-SCHEDUL-MAJOR-REQMT.
      *    SQL-SCHEDUL-MIN-GPA
      *    MOVE COMM-VIS-JOB-LOCATION TO SQL-SCHEDUL-JOB-LOCATION.
      *    MOVE COMM-VIS-GRAD-DATE    TO SQL-SCHEDUL-GRAD-DATE.
      *    MOVE COMM-VIS-BID-DATE     TO SQL-SCHEDUL-BID-DATE.
      *    MOVE COMM-VIS-NOTES        TO SQL-SCHEDUL-NOTES.


       EXIT-GET-DFHCOMMAREA.

       90200-PUT-DFHCOMMAREA SECTION.
      *    Insert code to write the COMMAREA here.


       EXIT-PUT-DFHCOMMAREA.

       99000-ABNORMAL-TERMINATION SECTION.
      *    All abnormal terminations handled from here.
       99200-DB-ABEND.                                                  0
      ************************************************************** 00623000
      * THE FOLLOWING ROUTINE PRINTS THE SQLCA STRUCTURE:            * 00624000
      *                                                              * 00625000
      *  - SQLCODE = SQL RETURN CODE                                 * 00626000
      *  - SQLERRM = SQL ERROR MESSAGE                               * 00627000
      *  - SQLERRP = MODULE DETECTING ERROR                          * 00628000
      *  - SQLERRD = INTERNAL ERROR VALUES                           * 00629000
      *  - SQLWARN = SQL WARNING STRUCTURE                           * 00630000
      *                                                              * 00631000
      ************************************************************** 00632000
           DISPLAY '*******************************' UPON CONSOLE.   00633000
           DISPLAY '* PROGRAM ERROR ROUTINE ENTERED *' UPON CONSOLE. 00634000
           DISPLAY '* CHECK SYSPRINT FOR ERROR CODES*' UPON CONSOLE. 00635000
           DISPLAY '* CHANGES WILL BE BACKED OUT    *' UPON CONSOLE. 00636000
           DISPLAY '*******************************' UPON CONSOLE.   00637000
           MOVE SQLCODE TO DECODED-SQLCODE.                          00638000
           DISPLAY 'PROGRAM ERROR ROUTINE ENTERED'.                  00639000
           DISPLAY '       *******'.                                 00640000
           DISPLAY 'A PROBLEM HAS BEEN DETECTED IN THE '.            00641000
           DISPLAY STEP-INDICATOR, ' PARAGRAPH.'.                    00642000
           DISPLAY 'THE FOLLOWING ERROR CODES SHOULD AID YOU IN'.    00643000
           DISPLAY 'PROBLEM DETERMINATION OF THE SQL STATEMENT.'.    00644000
           DISPLAY '       *******'.                                 00645000
           DISPLAY 'SQLCODE : ' DECODED-SQLCODE.                     00646000
           DISPLAY 'SQLERRM : ' SQLERRMC.                            00647000
           DISPLAY 'SQLERRP : ' SQLERRP.                             00648000
           PERFORM ERRD VARYING INDX2 FROM 1 BY 1 UNTIL INDX2 = 7.   00649000
           IF SQLWARN0 NOT EQUAL 'W'                                 00650000
               THEN GO TO 99000-BACKOUT,                             00651000
               ELSE DISPLAY 'SQLWARN0: ' SQLWARN0,                   00652000
                    DISPLAY 'SQLWARN1: ' SQLWARN1,                   00653000
                    DISPLAY 'SQLWARN2: ' SQLWARN2,                   00654000
                    DISPLAY 'SQLWARN3: ' SQLWARN3,                   00655000
                    DISPLAY 'SQLWARN4: ' SQLWARN4,                   00656000
                    DISPLAY 'SQLWARN5: ' SQLWARN5,                   00657000
                    DISPLAY 'SQLWARN6: ' SQLWARN6,                   00658000
                    DISPLAY 'SQLWARN7: ' SQLWARN7,                   00659000
                    DISPLAY 'SQLWARN8: ' SQLWARN8,                   00660000
                    DISPLAY 'SQLWARN9: ' SQLWARN9,                   00661000
                    DISPLAY 'SQLWARNA: ' SQLWARNA,                   00662000
                    GO TO 99000-BACKOUT.                             00663000

       ERRD.
           MOVE SQLERRD (INDX2) TO DECODED-SQLERRD (INDX2).           0
           DISPLAY 'SQLERRD', INDX2, ': ', DECODED-SQLERRD (INDX2).  00666000
       EXIT-ERROR-ABEND.                                              0621000

       99000-BACKOUT SECTION.                                        00667000
      ************************************************************** 00668000
      * 'WHENEVER' RESET TO 'CONTINUE' IN THE EVENT THAT THE ROLLBACK * 00669000
      * WORK STATEMENT FAILS TO AVOID LOOP IN ERROR ROUTINE.         * 00670000
      ************************************************************** 00671000
                                                                      00672000
           MOVE 'ABEND - BACKING OUT' TO STEP-INDICATOR.               0674000
                                                                      00675000
           EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.              00676000
           EXEC SQL ROLLBACK WORK END-EXEC.                           00677000
           STOP RUN.
```

4) Source Code of Module: CPPO0029

```
*
*****************************************************************
*                                                               *
* BID PROCESSING PROGRAM - - CPPO0029                           *
*                                                               *
* DISPLAYS "RUN BID PROCESSING" SCREEN WHICH DISPLAYS HISTORY   *
* ABOUT THE LAST EXECUTION OF THE PROGRAM. USER HAS THE         *
* OPPORTUNITY TO EITHER PROCEED WITH EXECUTION OR CANCEL.       *
* INITIATES THE BID PROCESSING/INTERVIEW AUCTIONING SESSION     *
* WHICH PROCESSES ALL BIDS FOR EACH EMPLOYER WHOSE BID DATE     *
* IS IDENTICAL TO THE DATE WHICH THE PROGRAM IS EXECUTED.       *
* IT WILL DETERMINE THE WINNING BID GIVEN THE NUMBER OF         *
* INTERVIEW SLOTS THAT ARE OPEN AND ALL OF THE STUDENTS BIDS.   *
* USING THE WINNING BID VALUE, THE STUDENT WITH WINNING         *
* BIDS ARE NOTIFIED. UPDATES THE STUDENT BID POINT TOTAL        *
* FOLLOWING TWO RULES:(******REMEMBER******==> THE AMOUNT       *
* THE STUDENT BID WAS SUBTRACTED FROM THEIR TOTAL AT THE        *
* TIME THEY PLACED THE BID)                                     *
*                                                               *
*     1. FOR STUDENTS WITH LOSING BIDS - ADD THE TOTAL AMOUNT   *
*        OF THEIR BID BACK TO THEIR TOTAL                       *
*     2. FOR STUDENTS WITH WINNING BIDS - SUBTRACT THE AMOUNT   *
*        OF THE WINNING BID FROM THE STUDENTS BID AND ADD       *
*        THE DIFFERENCE BACK TO THEIR TOTAL                     *
*                                                               *
* ONCE THE WINNING BIDS ARE DETERMINED, A WINNING BID REPORT    *
* IS GENERATED.                                                 *
*                                                               *
* LIST OF SCREENS: 2.4.6 RUN BID PROCESSING                     *
*                                                               *
* LIST OF TABLES:  STUDENT TABLE                                *
*                  SCHEDULE TABLE                               *
*                  BID TABLE                                    *
*                  EMPLOYER TABLE                               *
*                                                               *
* TEAM MEMBERS:                                                 *
*     1. GLENN JARVIS                                           *
*     2. MIKE DAVIS                                             *
*     3. RICH URBANCIC                                          *
*     4. SCOTT ENGLERT                                          *
*     MODIFIED 10/92 STEVE PETER; EMBEDDED SQL & LOGIC REVAMP.  *
*****************************************************************

 IDENTIFICATION DIVISION.
 PROGRAM-ID.   CPPO0029.
 AUTHOR.       GLENN JARVIS & STEVE PETER.

 eject
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
     C01 IS TOP-OF-PAGE.

 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT AUDIT-TRAIL ASSIGN TO AUDIT.
     SELECT OUTPUT-FILE ASSIGN TO BIDRPT.


 DATA DIVISION.
 FILE SECTION.

 FD AUDIT-TRAIL
     LABEL RECORD IS STANDARD
     RECORDING MODE IS F
```

```
     RECORD CONTAINS 33 CHARACTERS.

 01 AUDIT-RECORD.
     05 INPUT-NAME           PIC X(20).
     05 INPUT-DATE           PIC X(8).
     05 INPUT-TIME           PIC X(5).

 FD OUTPUT-FILE
     LABEL RECORD IS STANDARD
     RECORDING MODE IS F
     RECORD CONTAINS 132 CHARACTERS.

 01 OUTPUT-RECORD            PIC X(132).


 WORKING-STORAGE SECTION.
     EXEC SQL INCLUDE SQLCA END-EXEC.                          00100000
*    Additional variables for abnormal termination.
 01  DECODED-SQLCODE         PIC ---------999.                 00107000
                                                               00108000
                                                               00109000
 01  ARRAY-SQLERRD.
     02 DECODED-SQLERRD      PIC ---------999 OCCURS 6 TIMES.  00110000
 01  INDX2                   PIC S9(1) SYNC USAGE IS COMP.     00111000
 01  INDXPIC                 PIC ZZZ9.                         00112000
 eject                                                         00101000
***************************************************************** 00102000
*   PROGRAM VARIABLE DECLARTION SECTION                         * 00103000
***************************************************************** 00104000
 01  STEP-INDICATOR          PIC X(30) VALUE 'PREP-SCREEN'.      001050
 01  DB-STATUS-INDICATOR     PIC X(8)  VALUE SPACES.
 01  EDIT-STATUS-INDICATOR   PIC X(8)  VALUE SPACES.
 01  THIS-PROGRAM-ID         PIC X(8)  VALUE 'CPPO0029'.
 01  SCRN-NAME               PIC X(8)  VALUE 'SCRN2460'.
 01  SCRN-RETCODE            PIC S9(7) COMP.
 01  SCRN-FIELD-NAME         PIC X(7).
 01  FIELDNAME               PIC X(7)  VALUE 'PGMID'.
 01  BID-POINTS              PIC 9(5).
 01  WS-NUMBER               PIC 9(5).
 01  WS-SSN                  PIC 9(9).
 01  POINTS                  PIC 9(5).
 01  WS-POINTS-BID           PIC 9(5).
 01  STUDENT-POINTS          PIC 9(5).
*01  EMP-NO                  PIC X(3).
*01  SCH-NO                  PIC X(2).
 01  BIDS-EXCEED-SLOTS       PIC X(1).
 01  VISIT-NDX               PIC 9(2).
 01  VISIT-NDX2              PIC 9(2).
 01  BID-NDX                 PIC 9(3).
 01  LOSE-NDX                PIC 9(3).
 01  SAVED-VISIT-NDX         PIC 9(2).
 01  SAVED-BID-NDX           PIC 9(3).
 01  SAVED-LOSE-NDX          PIC 9(3).
 01  TABLE-ENTRIES           PIC 9(5).
 01  WS-RECORDS              PIC X.
     88 NO-MORE-RECORDS      VALUE 'Y'.
 01  RETURNED-KEY            PIC X(8).
  EJECT

*****************************************************************
*                                                               *
*    TABLE DEFINITIONS                                          *
*                                                               *
*****************************************************************
 01 LOSER-TABLE.
     05 SQL-RECORD OCCURS 800 TIMES.
*    05 SQL-RECORD OCCURS 800 TIMES INDEXED BY LOSE-NDX.
```

```
           10 LOSE-TABLE-SSN                    PIC X(9).
           10 LOSE-TABLE-POINTS-BID              PIC 9(5).

       01 BID-TABLE.
           05 SQL-RECORD OCCURS 400 TIMES.
      *    05 SQL-RECORD OCCURS 400 TIMES INDEXED BY BID-NDX.
              10 BID-TABLE-STATUS                PIC X(8).
              10 BID-TABLE-EMP-NO                PIC X(5).
              10 BID-TABLE-POINTS-BID            PIC 9(5).
              10 BID-TABLE-SCH-NO                PIC X(2).
              10 BID-TABLE-SSN                   PIC X(9).
              10 BID-TABLE-DATE-OF-BID           PIC X(10).
              10 BID-TABLE-TIME-OF-BID           PIC X(8).

       01 VISIT-TABLE.
           05 SQL-RECORD OCCURS 50 TIMES.
      *    05 SQL-RECORD OCCURS 50 TIMES INDEXED BY VISIT-NDX.
              10 VIS-TABLE-EMP-NO                PIC X(5).
              10 VIS-TABLE-SCH-NO                PIC X(2).
              10 VIS-TABLE-NUM-OPEN-SLOTS        PIC 9(5).


       01 WS-DATE.
           05 WS-YY                  PIC XX.
           05 WS-MON                 PIC XX.
           05 WS-DD                  PIC XX.

       01 WS-TIME.
           05 WS-HR                  PIC XX.
           05 WS-MM                  PIC XX.
           05 WS-SS                  PIC XX.
       EJECT
      ****************************************************************
      *                                                              *
      *    PARAMETERS SENT TO THE SCREEN PROGRAM                     *
      *                                                              *
      ****************************************************************

       01 WS-DATA-FIELDS.
           05 WS-SCRN-NUMBER            PIC 9(7) COMP.
           05 WS-PGMID                  PIC X(8) VALUE 'CPP00029'.
           05 WS-SCRNID                 PIC X(8) VALUE ' 2-4-6  '.
           05 WS-SCRNDATE.
              10 WS-SCRN-MON            PIC X(2).
              10 FILLER                 PIC X    VALUE '-'.
              10 WS-SCRN-DD             PIC X(2).
              10 FILLER                 PIC X    VALUE '-'.
              10 WS-SCRN-YY             PIC X(2).
           05 WS-SCRNTIME.
              10 WS-SCRN-HH             PIC X(2).
              10 FILLER                 PIC X    VALUE ':'.
              10 WS-SCRN-MM             PIC X(2).
              10 FILLER                 PIC X    VALUE ':'.
              10 WS-SCRN-SS             PIC X(2).
           05 WS-RUNDATE.
              10  WS-RUN-CENT           PIC X(2).
              10  WS-RUN-YR             PIC X(2).
              10  WS-RUN-MO             PIC X(2).
              10  WS-RUN-DY             PIC X(2).
           05 WS-RUNTIME                PIC X(5).
           05 WS-EXENAME                PIC X(21).
           05 WS-SYSMSG                 PIC X(71).
           05 WS-SCRNKEY                PIC X(8).
       EJECT
      ****************************************************************
      *
```

```
      *    REPORT HEADINGS                                           *
      *                                                              *
      ****************************************************************
       01  HEADING-1.
           05 FILLER                    PIC X(19)
              VALUE 'BID RESULTS REPORT '.
           05 RPT-DATE                  PIC X(10).
           05 FILLER                    PIC X(14)      VALUE SPACES.
           05 FILLER                    PIC X(89)  value spaces.

       01  HEADING-2.
           05 FILLER                    PIC X(26)
              VALUE 'EMPLOYER SCHEDULE NUMBER: '.
           05 RPT-EMP-NO                PIC X(5).
           05 FILLER                    PIC X          VALUE '-'.
           05 RPT-SCH-NO                PIC X(2).
           05 FILLER                    PIC X(02) VALUE SPACES.
           05 FILLER                    PIC X(15)
              VALUE 'EMPLOYER NAME: '.
           05 RPT-EMP-NAME              PIC X(30).
           05 FILLER                    PIC X(53)    VALUE SPACES.

       01  HEADING-3.
           05 FILLER                    PIC X(03) VALUE SPACES.
           05 FILLER                    PIC X(03) VALUE 'SSN'.
           05 FILLER                    PIC X(14) VALUE SPACES.
           05 FILLER                    PIC X(04) VALUE 'NAME'.
           05 FILLER                    PIC X(14) VALUE SPACES.
           05 FILLER                    PIC X(07) VALUE 'AMT BID'.
           05 FILLER                    PIC X(02) VALUE SPACES.
           05 FILLER                    PIC X(11) VALUE 'WINNING BID'.
           05 FILLER                    PIC X(74) VALUE SPACES.

       01  HEADING-UNDER.
           05 FILLER                       PIC X(48)
              VALUE '------------------------------------------------'.
           05 FILLER                       PIC X(20)
              VALUE '--------------------'.
           05 FILLER                       PIC X(72)  VALUE SPACES.

       01 RPT-BLANK-LINE                   PIC X(132) VALUE SPACES.
       EJECT
      ****************************************************************
      *                                                              *
      *    REPORT DATA LINE -- WINNING BID REPORT                    *
      *                                                              *
      ****************************************************************
       01  RPT-DATA.
           05 RPT-STU-SSN                  PIC X(9).
           05 FILLER                       PIC X(5)   VALUE SPACES.
           05 RPT-STU-NAME                 PIC X(20).
           05 FILLER                       PIC X(5)   VALUE SPACES.
           05 RPT-STU-AMT-BID              PIC 9(5).
           05 FILLER                       PIC X(5)   VALUE SPACES.
           05 RPT-WINNING-BID              PIC 9(5).

       eject
           EXEC SQL BEGIN DECLARE SECTION END-EXEC.                    EMP00010
       EJECT                                                           EMP00020
      *host variable used in "where" portion of query on BID Table.
       01  WS-SQL-DATE                     PIC X(10).
       01  SQL-BID-MARGIN                  PIC S9(4) COMP.
      ****************************************************************
      *                                                              *
      *    SQL-EMPLOYER ENTITY                                       *
      *
```

```
*****************************************************************
 01  SQL-EMP-ADDRESS                 PIC X(20).                  EMP00040
 01  SQL-EMP-CITY                    PIC X(20).                  EMP00050
 01  SQL-EMP-CONTACT                 PIC X(20).                  EMP00060
 77  SQL-EMP-EMP-NO                  PIC X(05).                  EMP00070
 01  SQL-EMP-NAME                    PIC X(30).                  EMP00080
 01  SQL-EMP-PHONE                   PIC X(10).                  EMP00090
 01  SQL-EMP-STATE                   PIC X(02).                  EMP00100
 01  SQL-EMP-ZIP                     PIC X(05).                  EMP00110
                                                                 EMP00120
     EXEC SQL END DECLARE SECTION  END-EXEC.                     EMP00130
                                                                 EMP00140
 01  EMP-RECORD.                                                 EMP00150
     10  EMP-ADDRESS                 PIC X(20).                  EMP00160
     10  EMP-CITY                    PIC X(20).                  EMP00170
     10  EMP-CONTACT                 PIC X(20).                  EMP00180
     10  EMP-EMP-NO                  PIC X(05).                  EMP00190
     10  EMP-NAME                    PIC X(30).                  EMP00200
     10  EMP-PHONE                   PIC X(10).                  EMP00210
     10  EMP-STATE                   PIC X(02).                  EMP00220
     10  EMP-ZIP                     PIC X(05).                  EMP00230

     EXEC SQL BEGIN DECLARE SECTION END-EXEC.
 EJECT
*****************************************************************
*                                                               *
*    SQL-BID ENTITY                                             *
*                                                               *
*****************************************************************
 01  SQL-BID-SSN                     PIC X(09).
 01  SQL-BID-EMP-NO                  PIC X(05).
 01  SQL-BID-SCH-NO                  PIC X(02).
 77  SQL-BID-POINTS-BID              PIC S9(4) COMP.
 01  SQL-BID-STATUS                  PIC X(08).
 01  SQL-BID-DATE-OF-BID             PIC X(10).
 01  SQL-BID-TIME-OF-BID             PIC X(8).

     EXEC SQL END DECLARE SECTION  END-EXEC.

 01  BID-RECORD.
     10  BID-SSN                     PIC X(09).
     10  BID-EMP-NO                  PIC X(05).
     10  BID-SCH-NO                  PIC X(02).
     10  BID-POINTS-BID              PIC 9(4).
     10  BID-STATUS                  PIC X(08).
     10  BID-DATE-OF-BID             PIC X(10).
     10  BID-TIME-OF-BID             PIC X(8).
     EXEC SQL BEGIN DECLARE SECTION END-EXEC.                    STU00010
 EJECT
*****************************************************************
*                                                               *
*    SQL-STUDENT ENTITY                                         *
*                                                               *
*****************************************************************
 01  SQL-STU-SQL-STU-SSN             PIC X(09).                  STU00040
 01  SQL-STU-LAST-NAME               PIC X(20).                  STU00050
 01  SQL-STU-FIRST-NAME              PIC X(15).                  STU00060
 01  SQL-STU-M-INITIAL               PIC X(01).                  STU00070
                                                                 STU00080
*    ADDRESS                                                    STU00090
 01  SQL-STU-PRES-ADDR               PIC X(20).                  STU00100
 01  SQL-STU-PRES-CITY               PIC X(20).                  STU00110
 01  SQL-STU-PRES-STATE              PIC X(02).                  STU00120
 01  SQL-STU-PRES-ZIP                PIC X(05).                  STU00130
 01  SQL-STU-PRES-PHONE              PIC X(10).                  STU00140
 01  SQL-STU-PERM-ADDR               PIC X(20).                  STU00150
```

```
 01  SQL-STU-PERM-CITY               PIC X(20).                  STU00160
 01  SQL-STU-PERM-STATE              PIC X(02).                  STU00170
 01  SQL-STU-PERM-ZIP                PIC X(05).                  STU00180
 01  SQL-STU-PERM-PHONE              PIC X(10).                  STU00190
                                                                 STU00200
*    DEMOGRAPHICS                                               STU00210
 01  SQL-STU-SEX                     PIC X(01).                  STU00220
 01  SQL-STU-CITIZENSHIP             PIC X(01).                  STU00230
                                                                 STU00240
*    CAREER INTEREST DATA                                       STU00250
 01  SQL-STU-CAREER-INT              PIC X(11).                  STU00260
 01  SQL-STU-GEOGRAPH-PREF           PIC X(10).                  STU00270
 01  SQL-STU-WORK-PREF               PIC X(20).                  STU00280
 01  SQL-STU-DATE-AVAIL              PIC X(10).                  STU00290
                                                                 STU00300
*    ACADEMIC RECORD                                            STU00310
 01  SQL-STU-DIVISION                PIC X(01).                  STU00320
 01  SQL-STU-MAJOR-1                 PIC X(03).                  STU00330
 01  SQL-STU-MAJOR-2                 PIC X(03).                  STU00340
 01  SQL-STU-MINOR                   PIC X(03).                  STU00350
 01  SQL-STU-CONCENTRATION           PIC X(20).                  STU00360
 01  SQL-STU-DEGREE                  PIC X(01).                  STU00370
 01  SQL-STU-GRAD-DATE               PIC X(04).                  STU00380
 77  SQL-STU-GPA-OVERALL             PIC S9V99 COMP-3.           STU00390
 77  SQL-STU-GPA-MAJOR-1             PIC S9V99 COMP-3.           STU00400
 77  SQL-STU-GPA-MAJOR-2             PIC S9V99 COMP-3.           STU00410
 77  SQL-STU-GPA-FRESHMAN            PIC S9V99 COMP-3.           STU00420
 77  SQL-STU-GPA-SOPHOMORE           PIC S9V99 COMP-3.           STU00430
 77  SQL-STU-GPA-JUNIOR              PIC S9V99 COMP-3.           STU00440
 77  SQL-STU-GPA-SENIOR              PIC S9V99 COMP-3.           STU00450
 77  SQL-STU-GPA-GRADUATE            PIC S9V99 COMP-3.           STU00460
                                                                 STU00470
*    CPPO DATA                                                  STU00480
 77  SQL-STU-BID-POINTS              PIC S9(4) COMP.             STU00490
                                                                 STU00500
*    EMPLOYMENT HISTORY                                         STU00510
 01  SQL-STU-PERCENT-EARNED          PIC S9(4) COMP.             STU00520
 01  SQL-STU-EMPLOYER-1              PIC X(20).                  STU00530
 01  SQL-STU-LOCATION-1              PIC X(20).                  STU00540
 01  SQL-STU-JOB-CATAGORY-1          PIC X(03).                  STU00550
 01  SQL-STU-FROM-DATE-1             PIC X(04).                  STU00560
 01  SQL-STU-TO-DATE-1               PIC X(04).                  STU00570
 01  SQL-STU-WORK-DESC-1             PIC X(40).                  STU00580
 01  SQL-STU-EMPLOYER-2              PIC X(20).                  STU00590
 01  SQL-STU-LOCATION-2              PIC X(20).                  STU00600
 01  SQL-STU-JOB-CATAGORY-2          PIC X(03).                  STU00610
 01  SQL-STU-FROM-DATE-2             PIC X(04).                  STU00620
 01  SQL-STU-TO-DATE-2               PIC X(04).                  STU00630
 01  SQL-STU-WORK-DESC-2             PIC X(40).                  STU00640
 01  SQL-STU-EMPLOYER-3              PIC X(20).                  STU00650
 01  SQL-STU-LOCATION-3              PIC X(20).                  STU00660
 01  SQL-STU-JOB-CATAGORY-3          PIC X(03).                  STU00670
 01  SQL-STU-FROM-DATE-3             PIC X(04).                  STU00680
 01  SQL-STU-TO-DATE-3               PIC X(04).                  STU00690
 01  SQL-STU-WORK-DESC-3             PIC X(40).                  STU00700
                                                                 STU00710
*    ACTIVITIES, HONORS, SKILLS                                 STU00720
 01  SQL-STU-ACTIVITIES              PIC X(240).                 STU00730
 01  SQL-STU-SPECIAL-SKILLS          PIC X(40).                  STU00740
                                                                 STU00750
*    TEACHING DATA                                              STU00760
 01  SQL-STU-TEACHER-CERT-1          PIC X(40).                  STU00770
 01  SQL-STU-TEACHER-CERT-2          PIC X(40).                  STU00780
 01  SQL-STU-INTERNSHIP              PIC X(40).                  STU00790
                                                                 STU00800
*    CPPO DATA                                                  STU00810
```

```
 01  SQL-STU-SUSPEND-STATUS              PIC X(08).

*    COLLEGES ATTENDED
 01  SQL-STU-OTHER-COLLEGE-1             PIC X(254).
 01  SQL-STU-OTHER-COLLEGE-2             PIC X(254).
     EXEC SQL END DECLARE SECTION END-EXEC.

 eject
 01  SQL-STUDENT.
*    IDENTIFICATION DATA
     10  STU-SQL-STU-SSN                 PIC X(09).
     10  STU-LAST-NAME                   PIC X(20).
     10  STU-FIRST-NAME                  PIC X(15).
     10  STU-M-INITIAL                   PIC X(01).

*    ADDRESS
     10  STU-PRES-ADDR                   PIC X(20).
     10  STU-PRES-CITY                   PIC X(20).
     10  STU-PRES-STATE                  PIC X(02).
     10  STU-PRES-ZIP                    PIC X(05).
     10  STU-PRES-PHONE                  PIC X(10).
     10  STU-PERM-ADDR                   PIC X(20).
     10  STU-PERM-CITY                   PIC X(20).
     10  STU-PERM-STATE                  PIC X(02).
     10  STU-PERM-ZIP                    PIC X(05).
     10  STU-PERM-PHONE                  PIC X(10).

*    DEMOGRAPHICS
     10  STU-SEX                         PIC X(01).
     10  STU-CITIZENSHIP                 PIC X(01).

*    CAREER INTEREST DATA
     10  STU-CAREER-INT                  PIC X(11).
     10  STU-GEOGRAPH-PREF               PIC X(10).
     10  STU-WORK-PREF                   PIC X(20).
     10  STU-DATE-AVAIL                  PIC X(10).

*    ACADEMIC RECORD
     10  STU-DIVISION                    PIC X(01).
     10  STU-MAJOR-1                     PIC X(03).
     10  STU-MAJOR-2                     PIC X(03).
     10  STU-MINOR                       PIC X(03).
     10  STU-CONCENTRATION               PIC X(20).
     10  STU-DEGREE                      PIC X(01).
     10  STU-GRAD-DATE                   PIC X(04).
     10  STU-GPA-OVERALL                 PIC S9V99.
     10  STU-GPA-MAJOR-1                 PIC S9V99.
     10  STU-GPA-MAJOR-2                 PIC S9V99.
     10  STU-GPA-FRESHMAN                PIC S9V99.
     10  STU-GPA-SOPHOMORE               PIC S9V99.
     10  STU-GPA-JUNIOR                  PIC S9V99.
     10  STU-GPA-SENIOR                  PIC S9V99.
     10  STU-GPA-GRADUATE                PIC S9V99.

*    CPPO DATA
     10  STU-BID-POINTS                  PIC S9(4).

*    EMPLOYMENT HISTORY
     10  STU-PERCENT-EARNED              PIC S9(4).
     10  STU-EMPLOYER-1                  PIC X(20).
     10  STU-LOCATION-1                  PIC X(20).
     10  STU-JOB-CATAGORY-1              PIC X(03).
     10  STU-FROM-DATE-1                 PIC X(04).
     10  STU-TO-DATE-1                   PIC X(04).
     10  STU-WORK-DESC-1                 PIC X(40).
     10  STU-EMPLOYER-2                  PIC X(20).
```

```
     10  STU-LOCATION-2                  PIC X(20).
     10  STU-JOB-CATAGORY-2              PIC X(03).
     10  STU-FROM-DATE-2                 PIC X(04).
     10  STU-TO-DATE-2                   PIC X(04).
     10  STU-WORK-DESC-2                 PIC X(40).
     10  STU-EMPLOYER-3                  PIC X(20).
     10  STU-LOCATION-3                  PIC X(20).
     10  STU-JOB-CATAGORY-3              PIC X(03).
     10  STU-FROM-DATE-3                 PIC X(04).
     10  STU-TO-DATE-3                   PIC X(04).
     10  STU-WORK-DESC-3                 PIC X(40).

*    ACTIVITIES, HONORS,SKILLS
     10  STU-ACTIVITIES                  PIC X(240).
     10  STU-SPECIAL-SKILLS              PIC X(40).

*    TEACHING DATA
     10  STU-TEACHER-CERT-1              PIC X(40).
     10  STU-TEACHER-CERT-2              PIC X(40).
     10  STU-INTERNSHIP                  PIC X(40).

*    CPPO DATA
     10  STU-SUSPEND-STATUS              PIC X(08).

*    COLLEGES ATTENDED
     10  STU-OTHER-COLLEGE-1             PIC X(254).
     10  STU-OTHER-COLLEGE-2             PIC X(254).

     EXEC SQL BEGIN DECLARE SECTION END-EXEC.

 EJECT
*****************************************************************
*                                                               *
*    SQL-SCHEDUL ENTITY                                         *
*                                                               *
*****************************************************************
 01  SQL-SCHEDUL-EMP-NO                  PIC X(5).
 01  SQL-SCHEDUL-SCH-NO                  PIC X(2).
 01  SQL-SCHEDUL-INT-DATE-1              PIC X(10).
 01  SQL-SCHEDUL-INT-DATE-2              PIC X(10).
 01  SQL-SCHEDUL-INT-DATE-3              PIC X(10).
 01  SQL-SCHEDUL-NUM-OPEN-SLOTS          PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-DAYS                 PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-ROOMS                PIC S9(4) COMP.
 01  SQL-SCHEDUL-INT-LENGTH              PIC S9(4) COMP.
 01  SQL-SCHEDUL-NO-INTERVIEWERS         PIC S9(4) COMP.
 01  SQL-SCHEDUL-CLOSED-INT              PIC X.
 01  SQL-SCHEDUL-DEGREE-REQMT            PIC X.
 01  SQL-SCHEDUL-MAJOR-REQMT             PIC X(50).
 01  SQL-SCHEDUL-GPA-MINIMUM             PIC S9V99 COMP-3.
 01  SQL-SCHEDUL-POSITION-AVAIL          PIC X(20).
 01  SQL-SCHEDUL-JOB-LOCATION            PIC X(20).
 01  SQL-SCHEDUL-CITIZENSHIP             PIC X.
 01  SQL-SCHEDUL-GRAD-DATE               PIC XXXX.
 01  SQL-SCHEDUL-LITERATURE              PIC X.
 01  SQL-SCHEDUL-VIDEO                   PIC X.
 01  SQL-SCHEDUL-PRE-NIGHT-LOCATION      PIC X(20).
 01  SQL-SCHEDUL-PRE-NIGHT-DATE          PIC X(10).
 01  SQL-SCHEDUL-PRE-NIGHT-TIME          PIC X(10).
 01  SQL-SCHEDUL-BID-DATE                PIC X(10).
 01  SQL-SCHEDUL-NOTES                   PIC X(20).
 01  SQL-SCHEDUL-POINTS-USED-TO-WIN      PIC S9(4) COMP.
 01  SQL-SCHEDUL-STATUS                  PIC X(8).

     EXEC SQL END DECLARE SECTION END-EXEC.
```

```
01 SCHEDUL-RECORD.
   10  SCHEDUL-EMP-NO                  PIC X(5).
   10  SCHEDUL-SCH-NO                  PIC X(2).
   10  SCHEDUL-INT-DATE-1              PIC X(10).
   10  SCHEDUL-INT-DATE-2              PIC X(10).
   10  SCHEDUL-INT-DATE-3              PIC X(10).
   10  SCHEDUL-NUM-OPEN-SLOTS          PIC 9(4).
   10  SCHEDUL-NO-DAYS                 PIC 9(4).
   10  SCHEDUL-NO-ROOMS                PIC 9(4).
   10  SCHEDUL-INT-LENGTH              PIC 9(4).
   10  SCHEDUL-NO-INTERVIEWERS         PIC 9(4).
   10  SCHEDUL-CLOSED-INT              PIC X.
   10  SCHEDUL-DEGREE-REQMT            PIC X.
   10  SCHEDUL-MAJOR-REQMT             PIC X(50).
   10  SCHEDUL-GPA-MINIMUM             PIC 9.99.
   10  SCHEDUL-POSITION-AVAIL          PIC X(20).
   10  SCHEDUL-JOB-LOCATION            PIC X(20).
   10  SCHEDUL-CITIZENSHIP             PIC X.
   10  SCHEDUL-GRAD-DATE               PIC XXXX.
   10  SCHEDUL-LITERATURE              PIC X.
   10  SCHEDUL-VIDEO                   PIC X.
   10  SCHEDUL-PRE-NIGHT-LOCATION      PIC X(20).
   10  SCHEDUL-PRE-NIGHT-DATE          PIC X(10).
   10  SCHEDUL-PRE-NIGHT-TIME          PIC X(10).
   10  SCHEDUL-BID-DATE                PIC X(10).
   10  SCHEDUL-NOTES                   PIC X(20).
   10  SCHEDUL-POINTS-USED-TO-WIN      PIC 9(4).

 EJECT
*****************************************************************
*                                                               *
*    SCREEN ATTRIBUTES                                          *
*                                                               *
*****************************************************************
    77 PROT                        PIC X(7) VALUE 'PROT   '.
    77 UNPROT                      PIC X(7) VALUE 'UNPROT '.
    77 NUMRIC                      PIC X(7) VALUE 'NUMERIC'.
    77 BRIGHT                      PIC X(7) VALUE 'BRIGHT '.
    77 DIM                         PIC X(7) VALUE 'DIM    '.
    77 DARK                        PIC X(7) VALUE 'DIM    '.
    77 SKIP                        PIC X(7) VALUE 'SKIP   '.
    77 NOSKIP                      PIC X(7) VALUE 'NOSKIP '.
    77 MDT                         PIC X(7) VALUE 'MDT    '.
    77 NOMDT                       PIC X(7) VALUE 'NOMDT  '.
 eject
 01 PREV-EMP-NO PIC X(5).
 01 PREV-SCH-NO PIC X(2).
 01 WS-LINE-NUMBER PIC 99.

 01 AUDIT-RECORD-EOF-FLAG PIC X VALUE 'N'.
 01 CPPO-STAFF-NAME PIC X(20) VALUE 'JOE CPPO '.
*************************************************************
 01  SCRN-NUMBER    PIC     9(7) COMP.

 01  SCRN-FIELDS.
     05 PGMID          PIC     X(8).
     05 SCRNID         PIC     X(8).
     05 SCRDATE        PIC     X(8).
     05 SCRTIME        PIC     X(8).
     05 RUNDATE        PIC     X(8).
     05 RUNTIME        PIC     X(5).
     05 EXENAME        PIC    X(21).
     05 SYSMSG         PIC    X(71).
 01  SCRN-KEY       PIC     X(8).
     88  PF1                        VALUE 'PF01'.
     88  PF2                        VALUE 'PF02'.
     88  PF3                        VALUE 'PF03'.
     88  PF4                        VALUE 'PF04'.
     88  PF5                        VALUE 'PF05'.
     88  PF6                        VALUE 'PF06'.
     88  PF7                        VALUE 'PF07'.
     88  PF8                        VALUE 'PF08'.
     88  PF9                        VALUE 'PF09'.
     88  PF10                       VALUE 'PF10'.
     88  PF11                       VALUE 'PF11'.
     88  PF12                       VALUE 'PF12'.
     88  RETURN-KEY                 VALUE 'RETURN'.
*****************************************************************
 eject                                                          00113000
 01  TERMINAL-MESSAGES.                                        00114000
     02  REC-NOT-FOUND      PIC X(80) VALUE                    00115000
        ' RECORD NOT FOUND - USE A DIFFERENT KEY.'.             00116000
     02  DUPLICATE-RECORD   PIC X(80) VALUE                    00117000
        ' ATTEMPT TO ADD A DUPLICATE RECORD WAS REJECTED.'.     00118000
 EJECT
 LINKAGE SECTION.

     COPY DFHCOMM.

 EJECT
*****************************************************************
*                                                               *
*    PROCEDURE DIVISION                                         *
*                                                               *
*****************************************************************
 PROCEDURE DIVISION USING DFHCOMMAREA.
     EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
     EXEC SQL WHENEVER SQLERROR GO TO 9200-DB-ABEND END-EXEC.
 EJECT
*****************************************************************
*                                                               *
*    0000-MAINLINE SECTION.                                     *
*                                                               *
*****************************************************************
 0000-MAINLINE SECTION.
     PERFORM 90100-GET-DFHCOMMAREA.
     MOVE 'PREP-SCREEN' TO STEP-INDICATOR.
     OPEN OUTPUT OUTPUT-FILE.
     CALL 'MLOAD' USING SCRN-NUMBER SCRN-RETCODE SCRN-NAME.
     IF SCRN-RETCODE NOT = 0 THEN DISPLAY 'SCREEN ERROR'.
     MOVE SCRN-NUMBER TO WS-SCRN-NUMBER.
     PERFORM UNTIL STEP-INDICATOR = 'XFER-CONTROL'
        EVALUATE TRUE
            WHEN STEP-INDICATOR = 'PREP-SCREEN'
                 PERFORM 1000-PREPARE-SCREEN-OUTPUT
            WHEN STEP-INDICATOR = 'DISP-SCREEN'
                 PERFORM 2000-DISPLAY-SCREEN
            WHEN STEP-INDICATOR = 'EVAL-PFKEYS'
                 PERFORM 3000-EVALUATE-PFKEYS
            WHEN OTHER
               DISPLAY 'STEP INDICATOR NOT SET'
               PERFORM 99000-BACKOUT
        END-EVALUATE
     END-PERFORM.
     PERFORM 3100-SAVE-LAST-PERSON-TO-RUN.
     PERFORM 4000-TRANSFER-CONTROL THRU EXIT-TRANSFER-CONTROL.
     EXEC SQL COMMIT WORK END-EXEC.
     CLOSE OUTPUT-FILE.
     GOBACK.
 EXIT-MAINLINE.
```

```
 EJECT
*****************************************************************
*                                                               *
*    0000-PREPARE-SCREEN-OUPUT SECTION.                         *
*                                                               *
*****************************************************************

 1000-PREPARE-SCREEN-OUTPUT SECTION.

     ACCEPT WS-TIME FROM TIME.
     ACCEPT WS-DATE FROM DATE.
*following line is for implementation purposes only.
*also take out similar move in procedure 3100-.
     MOVE '930113' TO WS-DATE.
     MOVE WS-YY TO WS-SCRN-YY.
     MOVE WS-MON TO WS-SCRN-MON.
     MOVE WS-DD TO WS-SCRN-DD.
     MOVE WS-HH TO WS-SCRN-HH.
     MOVE WS-MM TO WS-SCRN-MM.
     MOVE WS-SS TO WS-SCRN-SS.

     OPEN INPUT AUDIT-TRAIL.
     PERFORM 1500-FIND-WHO-USED-LAST
        UNTIL AUDIT-RECORD-EOF-FLAG = 'Y'.
     CLOSE AUDIT-TRAIL.
*obtain date from file read previously. Modified S.Peter 10/22/92.
     PERFORM 1550-OBTAIN-DATE-FOR-QUERY.

     MOVE THIS-PROGRAM-ID TO PGMID.
     MOVE SCRN-NAME TO SCRNID.
     MOVE WS-SCRNDATE TO SCRDATE.
     MOVE WS-SCRNTIME TO SCRTIME.
     MOVE WS-EXENAME TO EXENAME.

     STRING WS-RUN-MO '/' WS-RUN-DY '/' WS-RUN-YR
            DELIMITED BY SIZE INTO RUNDATE.

     MOVE WS-RUNTIME TO RUNTIME.

     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.

 EXIT-PREPARE-SCREEN-OUTPUT.
 eject
*****************************************************************
*                                                               *
*    1500-FIND-WHO-USED-LAST SECTION.                           *
*                                                               *
*****************************************************************
 1500-FIND-WHO-USED-LAST SECTION.
     READ AUDIT-TRAIL
        AT END MOVE 'Y' TO AUDIT-RECORD-EOF-FLAG.
     MOVE INPUT-NAME    TO WS-EXENAME.
     MOVE INPUT-DATE    TO WS-RUNDATE.
     MOVE INPUT-TIME    TO WS-RUNTIME.

*****************************************************************
*                                                               *
*    1550-OBTAIN-DATE-FOR-QUERY.                                *
*                                                               *
*****************************************************************
 1550-OBTAIN-DATE-FOR-QUERY.
*needed the following move to a variable that could be used
*as a host variable in a query in the BID Table.
* this date is yesterday's business date.

     STRING WS-RUN-CENT WS-RUN-YR '-' WS-RUN-MO '-'
```

```
                WS-RUN-DY DELIMITED BY SIZE
                INTO WS-SQL-DATE.
 eject
*****************************************************************
*                                                               *
*    2000-DISPLAY-SCREEN SECTION.                               *
*                                                               *
*****************************************************************

 2000-DISPLAY-SCREEN SECTION.
*    Replace with application specific call to XMENU.
*    There is also a scrn call stmt in pf4 EVALUATE stmt.

     CALL 'MPCUR' USING SCRN-NUMBER, SCRN-RETCODE,
                                                FIELDNAME.
     CALL 'SCRN2460' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.
     MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.
 EXIT-DISPLAY-SCREEN.

*****************************************************************
*                                                               *
*    3000-EVALUATE-PFKEYS SECTION.                              *
*                                                               *
*****************************************************************
 3000-EVALUATE-PFKEYS SECTION.
     EVALUATE TRUE
        WHEN PF1
             MOVE 'HELP SCREEN NOT COMPLETED YET' TO SYSMSG
             MOVE 'DISP-SCREEN' TO STEP-INDICATOR
        WHEN PF3
             MOVE 'XFER-CONTROL' TO STEP-INDICATOR
        WHEN PF4
             PERFORM 5000-PROCESS-SECTION
             MOVE '=======> BID PROCESSING COMPLETE <========'
                                  TO SYSMSG
             MOVE 'DISP-SCREEN' TO STEP-INDICATOR
        WHEN OTHER
             MOVE '**** INVALID PF KEY INPUT FOR THIS SCREEN ****'
                TO SYSMSG
             MOVE 'DISP-SCREEN' TO STEP-INDICATOR
     END-EVALUATE.
 EXIT-EVALUATE-PFKEYS.
     EXIT.
 eject
*****************************************************************
*                                                               *
*    3100-SAVE-LAST-PERSON-TO-RUN SECTION.                      *
*                                                               *
*****************************************************************

 3100-SAVE-LAST-PERSON-TO-RUN SECTION.
     OPEN EXTEND AUDIT-TRAIL.
     MOVE CPPO-STAFF-NAME TO INPUT-NAME.
     ACCEPT WS-TIME FROM TIME.
     ACCEPT WS-DATE FROM DATE.
*following line gets removed after development. S.Peter.
     move '930112' to ws-date.
     STRING '19' WS-YY WS-MON WS-DD DELIMITED BY SIZE
                  INTO INPUT-DATE.
     STRING WS-HH ':' WS-MM DELIMITED BY SIZE
                  INTO INPUT-TIME.
     WRITE AUDIT-RECORD.
     CLOSE AUDIT-TRAIL.
  eject
*****************************************************************
*                                                               *
```

```
*    4000-TRANSFER-CONTROL SECTION.                                *
*                                                                  *
********************************************************************

 4000-TRANSFER-CONTROL SECTION.
*    Populate those COMMAREA fields needed by next program.
*    PERFORM 90200-PUT-DFHCOMMAREA.
     MOVE PREV-PROGRAM-ID TO NEXT-PROGRAM-ID
     MOVE 'CPP00029' TO PREV-PROGRAM-ID.
     MOVE 0 TO RETURN-CODE.
 EXIT-TRANSFER-CONTROL.
     EXIT.
 eject
********************************************************************
*                                                                  *
*    5000-PROCESS-SECTION                                          *
*                                                                  *
********************************************************************
 5000-PROCESS-SECTION.
     PERFORM 90310-DECLARE-1 THRU 90310-EXIT.
     PERFORM 90320-DECLARE-2 THRU 90320-EXIT.
     PERFORM 90410-OPEN-1    THRU 90410-EXIT.
     PERFORM 90500-FETCH-AND-TABLE-VISIT VARYING VISIT-NDX FROM
             1 BY 1 UNTIL VISIT-NDX > 50 OR SQLCODE = 100.

     COMPUTE VISIT-NDX = VISIT-NDX - 2.
     MOVE VISIT-NDX TO SAVED-VISIT-NDX.

     MOVE 1 TO VISIT-NDX.
     PERFORM 6000-BID-LOOP UNTIL (VISIT-NDX > SAVED-VISIT-NDX).

     PERFORM 90910-CLOSE-1   THRU 90910-EXIT.
     PERFORM 90920-CLOSE-2   THRU 90920-EXIT.

 EXIT-PROCESS-SECTION.
     EXIT.

********************************************************************
*                                                                  *
*    6000-BID-LOOP.                                                *
*                                                                  *
********************************************************************

 6000-BID-LOOP.
     MOVE VIS-TABLE-EMP-NO(VISIT-NDX) TO SQL-SCHEDUL-EMP-NO.
     MOVE VIS-TABLE-SCH-NO(VISIT-NDX) TO SQL-SCHEDUL-SCH-NO.
     MOVE VIS-TABLE-NUM-OPEN-SLOTS(VISIT-NDX) TO
                    SQL-SCHEDUL-NUM-OPEN-SLOTS.
     MOVE VIS-TABLE-EMP-NO(VISIT-NDX) TO SQL-BID-EMP-NO.
     MOVE VIS-TABLE-SCH-NO(VISIT-NDX) TO SQL-BID-SCH-NO.
     MOVE 0 TO SQLCODE.
     PERFORM 90420-OPEN-2    THRU 90420-EXIT.
     PERFORM 90550-FETCH-TOP-BIDS-FOR-VISIT VARYING BID-NDX FROM
             1 BY 1 UNTIL BID-NDX > SQL-SCHEDUL-NUM-OPEN-SLOTS
             OR SQLCODE = 100.
* as for why i have to sub 1 here & 2 from visit-ndx,i do no know.
     COMPUTE BID-NDX = BID-NDX - 1.
     MOVE BID-NDX TO SAVED-BID-NDX.

     IF SQLCODE = 100 THEN
*    {need end of table when < num-open-slots to update status}
*    {of VISIT table to '#BID<SLT'.  S. Peter 10/24.92}
       MOVE 'N' TO BIDS-EXCEED-SLOTS
     ELSE
       IF (BID-NDX >= SQL-SCHEDUL-NUM-OPEN-SLOTS) THEN
         MOVE 'Y' TO BIDS-EXCEED-SLOTS.
```

```
*    {the last one pulled in from query is the charged pts for all.
     MOVE BID-TABLE-POINTS-BID(BID-NDX) TO
                                SQL-SCHEDUL-POINTS-USED-TO-WIN.
*going to need date & time for the add back loser points on tie.
     MOVE BID-TABLE-TIME-OF-BID(BID-NDX) TO SQL-BID-TIME-OF-BID.
     MOVE BID-TABLE-DATE-OF-BID(BID-NDX) TO SQL-BID-DATE-OF-BID.

     PERFORM 6100-HEADINGS.
     MOVE 1 TO BID-NDX.
     PERFORM 6150-DOWNLOAD-TABLE-TO-REPORT UNTIL
                                    (BID-NDX > SAVED-BID-NDX).


* note: check bid-ndx at end to see if it is one greater.
     IF (BIDS-EXCEED-SLOTS = 'N') THEN
        MOVE '#BID<SLT' TO SQL-SCHEDUL-STATUS
     ELSE IF (BIDS-EXCEED-SLOTS = 'Y') THEN
          MOVE '#BD>=SLT' TO SQL-SCHEDUL-STATUS
     ELSE
          MOVE 'ERROR   ' TO SQL-SCHEDUL-STATUS.

*    {update points used to win & status}
     PERFORM 90610-UPDATE-VISIT-RELATION.

     MOVE 1 TO BID-NDX.
*    {add back the bid winners difference in points.
     PERFORM 6200-ADJUST-STUDENT-POINTS UNTIL
                                    (BID-NDX > SAVED-BID-NDX).
*    {change the status field in the bid relation.- won, lost}
     PERFORM 6300-UPDATE-BID-RELATION.
     PERFORM 6400-ADD-BACK-LOSER-PTS.
     ADD 1 TO VISIT-NDX.


********************************************************************
*                                                                  *
*    6100-HEADINGS                                                 *
*                                                                  *
********************************************************************
 6100-HEADINGS.

     MOVE VIS-TABLE-EMP-NO(VISIT-NDX) TO RPT-EMP-NO
                                         SQL-EMP-EMP-NO.
     MOVE VIS-TABLE-SCH-NO(VISIT-NDX) TO RPT-SCH-NO
     MOVE WS-SQL-DATE TO RPT-DATE.
     PERFORM 90710-SELECT-EMPLOYER.
     MOVE SQL-EMP-NAME TO RPT-EMP-NAME.                          MF00210
     WRITE OUTPUT-RECORD FROM HEADING-1.
     WRITE OUTPUT-RECORD FROM RPT-BLANK-LINE.
     WRITE OUTPUT-RECORD FROM HEADING-2.
     WRITE OUTPUT-RECORD FROM RPT-BLANK-LINE.
     WRITE OUTPUT-RECORD FROM HEADING-3.
     WRITE OUTPUT-RECORD FROM RPT-BLANK-LINE.
     WRITE OUTPUT-RECORD FROM HEADING-UNDER.

 6100-EXIT.
     EXIT.
 EJECT

********************************************************************
*                                                                  *
*    6150-DOWNLOAD-TABLE-TO-REPORT.                                *
*                                                                  *
********************************************************************
 6150-DOWNLOAD-TABLE-TO-REPORT.
        MOVE BID-TABLE-SSN(BID-NDX) TO RPT-STU-SSN
```

```
                                        SQL-BID-SSN.
           PERFORM 90700-GET-STU-NAME.
           MOVE SQL-STU-LAST-NAME TO RPT-STU-NAME.
           MOVE BID-TABLE-POINTS-BID(BID-NDX) TO RPT-STU-AMT-BID.
           MOVE SQL-SCHEDUL-POINTS-USED-TO-WIN TO RPT-WINNING-BID.
           WRITE OUTPUT-RECORD FROM RPT-DATA.
           ADD 1 TO BID-NDX.

*****************************************************************
*                                                               *
*     6200-ADJUST-STUDENT-POINTS.                               *
*                                                               *
*****************************************************************
 6200-ADJUST-STUDENT-POINTS.
      MOVE BID-TABLE-SSN(BID-NDX) TO SQL-STU-SQL-STU-SSN.
      MOVE BID-TABLE-POINTS-BID(BID-NDX) TO SQL-BID-POINTS-BID.
      COMPUTE SQL-BID-MARGIN = (SQL-BID-POINTS-BID -
                                SQL-SCHEDUL-POINTS-USED-TO-WIN).
      PERFORM 90620-UPDATE-STUDENT-RELATION.
      ADD 1 TO BID-NDX.


*****************************************************************
*                                                               *
*     6300-UPDATE-BID-RELATION.                                 *
*                                                               *
*****************************************************************
 6300-UPDATE-BID-RELATION.
      PERFORM 90630-PROCESS-WON-STATUS.
      PERFORM 90640-PROCESS-LOST-STATUS.

*****************************************************************
*                                                               *
*     6400-ADD-BACK-LOSER-PTS.                                  *
*                                                               *
*****************************************************************
 6400-ADD-BACK-LOSER-PTS.

      PERFORM 90330-DECLARE-3.
      PERFORM 90430-OPEN-3.
      PERFORM 90575-FETCH-LOSERS VARYING LOSE-NDX FROM
              1 BY 1 UNTIL LOSE-NDX > 800 OR SQLCODE = 100.
      PERFORM 90930-CLOSE-3.
      COMPUTE LOSE-NDX = LOSE-NDX - 2.
      MOVE LOSE-NDX TO SAVED-LOSE-NDX.
      MOVE 1 TO LOSE-NDX.
      PERFORM 6450-ADD-BACK-PTS UNTIL (LOSE-NDX > SAVED-LOSE-NDX).

*****************************************************************
*                                                               *
*     6450-ADD-BACK-PTS.                                        *
*                                                               *
*****************************************************************
 6450-ADD-BACK-PTS.
      MOVE LOSE-TABLE-SSN(LOSE-NDX) TO SQL-STU-SQL-STU-SSN.
      MOVE LOSE-TABLE-POINTS-BID(LOSE-NDX) TO SQL-BID-POINTS-BID.
      PERFORM 90650-UPDATE-STUD-POINTS.
      ADD 1 TO LOSE-NDX.
*****************************************************************
*                                                               *
*     90100-GET-DFHCOMMAREA                                     *
*                                                               *
*****************************************************************
 90100-GET-DFHCOMMAREA SECTION.
*     Insert code to read the COMMAREA here.
 EXIT-GET-DFHCOMMAREA-EXIT.
```

```
 EJECT
*****************************************************************
*                                                               *
*     90200-PUT-DFHCOMMAREA                                     *
*                                                               *
*****************************************************************
 90200-PUT-DFHCOMMAREA SECTION.
*     Insert code to write the COMMAREA here.
*     and uncomment the perform stmt.
 EXIT-PUT-DFHCOMMAREA.
 EJECT

*****************************************************************
*                                                               *
*     90310-DECLARE-1                                           *
*                                                               *
*****************************************************************

 90310-DECLARE-1.

      EXEC SQL DECLARE C1 CURSOR FOR
               SELECT EMP_NO, SCH_NO, NUM_OPEN_SLOTS
               FROM VISIT
*  using previous school day (which is last day processed)
               WHERE BID_BY_DATE = :WS-SQL-DATE
               ORDER BY EMP_NO ASC, SCH_NO ASC
      END-EXEC.

 90310-EXIT.
     EXIT.
 EJECT
*****************************************************************
*                                                               *
*     90320-DECLARE-2                                           *
*                                                               *
*****************************************************************
 90320-DECLARE-2.

      EXEC SQL DECLARE C2 CURSOR FOR
               SELECT STATUS, EMP_NO,
               POINTS_BID, SSN, SCH_NO, DATE_OF_BID, TIME_OF_BID
               FROM BID
               WHERE
                DATE_OF_BID <= :WS-SQL-DATE
                AND STATUS = 'UNPROCED'
                AND EMP_NO = :SQL-SCHEDUL-EMP-NO
                AND SCH_NO = :SQL-SCHEDUL-SCH-NO
               ORDER BY POINTS_BID DESC,
                    DATE_OF_BID ASC, TIME_OF_BID ASC
      END-EXEC.
* note that it is the above order by statement that will deter-
*mine who the bid winners are.  A tie is decided by date and time.
*                                    S. Peter for Tom Schaber.
 90320-EXIT.
*****************************************************************
*                                                               *
*     90330-DECLARE-3                                           *
*                                                               *
*****************************************************************
 90330-DECLARE-3.

      EXEC SQL DECLARE C3 CURSOR FOR
               SELECT SSN, POINTS_BID
               FROM BID
               WHERE
                    STATUS = 'LOST'
```

```
              AND EMP_NO = :SQL-BID-EMP-NO
              AND SCH_NO = :SQL-BID-SCH-NO
             ORDER BY POINTS_BID DESC
     END-EXEC.

 EJECT
*****************************************************************
*                                                               *
*    90410-OPEN-1                                               *
*                                                               *
*****************************************************************
 90410-OPEN-1.

     EXEC SQL OPEN C1 END-EXEC.

 90410-EXIT.
 EJECT
*****************************************************************
*                                                               *
*    90420-OPEN-2                                               *
*                                                               *
*****************************************************************
 90420-OPEN-2.

     EXEC SQL OPEN C2 END-EXEC.

 90420-EXIT.
*****************************************************************
*                                                               *
*    90430-OPEN-3                                               *
*                                                               *
*****************************************************************
 90430-OPEN-3.

     EXEC SQL OPEN C3 END-EXEC.

 EJECT
*****************************************************************
*                                                               *
*    90500-FETCH-AND-TABLE-VISIT                                *
*                                                               *
*****************************************************************
 90500-FETCH-AND-TABLE-VISIT.

     EXEC SQL FETCH C1
        INTO :SQL-SCHEDUL-EMP-NO,
             :SQL-SCHEDUL-SCH-NO,
             :SQL-SCHEDUL-NUM-OPEN-SLOTS

      END-EXEC.


      MOVE SQL-SCHEDUL-EMP-NO      TO VIS-TABLE-EMP-NO(VISIT-NDX).
      MOVE SQL-SCHEDUL-SCH-NO      TO VIS-TABLE-SCH-NO(VISIT-NDX).
      MOVE SQL-SCHEDUL-NUM-OPEN-SLOTS TO
                           VIS-TABLE-NUM-OPEN-SLOTS(VISIT-NDX).
 90500-EXIT.
 EJECT
*****************************************************************
*                                                               *
*    90550-FETCH-TOP-BIDS-FOR-VISIT                             *
*                                                               *
*****************************************************************
 90550-FETCH-TOP-BIDS-FOR-VISIT.

     EXEC SQL FETCH C2
```

```
        INTO :SQL-BID-STATUS,
             :SQL-BID-EMP-NO,
             :SQL-BID-POINTS-BID,
             :SQL-BID-SSN,
             :SQL-BID-SCH-NO,
             :SQL-BID-DATE-OF-BID,
             :SQL-BID-TIME-OF-BID
      END-EXEC.

      MOVE SQL-BID-EMP-NO      TO BID-TABLE-EMP-NO(BID-NDX).
      MOVE SQL-BID-SCH-NO      TO BID-TABLE-SCH-NO(BID-NDX).
      MOVE SQL-BID-SSN         TO BID-TABLE-SSN(BID-NDX).
      MOVE SQL-BID-STATUS      TO BID-TABLE-STATUS(BID-NDX).
      MOVE SQL-BID-POINTS-BID  TO BID-TABLE-POINTS-BID(BID-NDX).
      MOVE SQL-BID-DATE-OF-BID TO BID-TABLE-DATE-OF-BID(BID-NDX).
      MOVE SQL-BID-TIME-OF-BID TO BID-TABLE-TIME-OF-BID(BID-NDX).

 90550-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*    90575-FETCH-LOSERS                                         *
*                                                               *
*****************************************************************
 90575-FETCH-LOSERS.

     EXEC SQL FETCH C3
        INTO :SQL-BID-SSN,
             :SQL-BID-POINTS-BID

      END-EXEC.

      MOVE SQL-BID-SSN        TO LOSE-TABLE-SSN(LOSE-NDX).
      MOVE SQL-BID-POINTS-BID TO LOSE-TABLE-POINTS-BID(LOSE-NDX).

 EJECT
*****************************************************************
*                                                               *
*    90610-UPDATE-VISIT-RELATION.                               *
*                                                               *
*****************************************************************
 90610-UPDATE-VISIT-RELATION.

            EXEC SQL
                UPDATE VISIT
                SET POINTS_USED_TO_WIN =
                   :SQL-SCHEDUL-POINTS-USED-TO-WIN,
                    STATUS = :SQL-SCHEDUL-STATUS
                WHERE EMP_NO = :SQL-SCHEDUL-EMP-NO
                AND   SCH_NO = :SQL-SCHEDUL-SCH-NO
            END-EXEC.

 90610-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*    90620-UPDATE-STUDENT-RELATION.                             *
*                                                               *
*****************************************************************

 90620-UPDATE-STUDENT-RELATION.

            EXEC SQL
                UPDATE STUDENT
                SET BID_POINTS_AVAIL =
                          BID_POINTS_AVAIL + :SQL-BID-MARGIN
```

```
                WHERE SSN = :SQL-STU-SQL-STU-SSN
            END-EXEC.

 90620-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*     90630-PROCESS-WON-STATUS.                                 *
*                                                               *
*****************************************************************
 90630-PROCESS-WON-STATUS.

     EXEC SQL
        UPDATE BID
        SET STATUS = 'WON'
        WHERE     EMP_NO = :SQL-BID-EMP-NO
           AND    SCH_NO = :SQL-BID-SCH-NO
           AND    STATUS = 'UNPROCED'
           AND    POINTS_BID > :SQL-SCHEDUL-POINTS-USED-TO-WIN
     END-EXEC.

     EXEC SQL
        UPDATE BID
        SET STATUS = 'WON'
        WHERE     EMP_NO = :SQL-BID-EMP-NO
           AND    SCH_NO = :SQL-BID-SCH-NO
           AND    STATUS = 'UNPROCED'
           AND    POINTS_BID = :SQL-SCHEDUL-POINTS-USED-TO-WIN
           AND    DATE_OF_BID < :SQL-BID-DATE-OF-BID
     END-EXEC.

     EXEC SQL
        UPDATE BID
        SET STATUS = 'WON'
        WHERE     EMP_NO = :SQL-BID-EMP-NO
           AND    SCH_NO = :SQL-BID-SCH-NO
           AND    STATUS = 'UNPROCED'
           AND    POINTS_BID = :SQL-SCHEDUL-POINTS-USED-TO-WIN
           AND    DATE_OF_BID = :SQL-BID-DATE-OF-BID
           AND    TIME_OF_BID <= :SQL-BID-TIME-OF-BID
     END-EXEC.


 90630-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*     90640-PROCESS-LOST-STATUS                                 *
*                                                               *
*****************************************************************
 90640-PROCESS-LOST-STATUS.

     EXEC SQL
          UPDATE BID
          SET STATUS = 'LOST'
           WHERE EMP_NO = :SQL-BID-EMP-NO
           AND   SCH_NO = :SQL-BID-SCH-NO
           AND   STATUS = 'UNPROCED'
           AND   POINTS_BID < :SQL-SCHEDUL-POINTS-USED-TO-WIN
     END-EXEC.

     EXEC SQL
          UPDATE BID
          SET STATUS = 'LOST'
           WHERE EMP_NO = :SQL-BID-EMP-NO
           AND   SCH_NO = :SQL-BID-SCH-NO
```

```
           AND   STATUS = 'UNPROCED'
           AND   POINTS_BID = :SQL-SCHEDUL-POINTS-USED-TO-WIN
           AND   DATE_OF_BID > :SQL-BID-DATE-OF-BID
     END-EXEC.

     EXEC SQL
          UPDATE BID
          SET STATUS = 'LOST'
           WHERE EMP_NO = :SQL-BID-EMP-NO
           AND   SCH_NO = :SQL-BID-SCH-NO
           AND   STATUS = 'UNPROCED'
           AND   POINTS_BID = :SQL-SCHEDUL-POINTS-USED-TO-WIN
           AND   DATE_OF_BID = :SQL-BID-DATE-OF-BID
           AND   TIME_OF_BID > :SQL-BID-TIME-OF-BID
     END-EXEC.

 90640-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*     90650-UPDATE-STUD-POINTS.                                 *
*                                                               *
*****************************************************************
 90650-UPDATE-STUD-POINTS.

     EXEC SQL
          UPDATE STUDENT
          SET BID_POINTS_AVAIL =
              BID_POINTS_AVAIL + :SQL-BID-POINTS-BID
          WHERE SSN = :SQL-STU-SQL-STU-SSN
     END-EXEC.

 90650-EXIT.
     EXIT.
*****************************************************************
*                                                               *
*     90700-GET-STU-NAME.                                       *
*                                                               *
*****************************************************************
 90700-GET-STU-NAME.

          EXEC SQL
              SELECT LAST_NAME
              INTO :SQL-STU-LAST-NAME
              FROM STUDENT
              WHERE SSN = :SQL-BID-SSN
          END-EXEC.

*****************************************************************
*                                                               *
*     90710-SELECT-EMPLOYER                                     *
*                                                               *
*****************************************************************
 90710-SELECT-EMPLOYER.                                          EMP00020
     EXEC SQL                                                     MP00030
         SELECT NAME INTO                                        EMP00080
          :SQL-EMP-NAME                                          EMP00120
         FROM EMPLOYER                                           EMP00130
         WHERE EMP_NO =:SQL-EMP-EMP-NO                           EMP00140
     END-EXEC.
 90710-EXIT.
     EXIT.
 EJECT
*****************************************************************
*                                                               *
*     90910-CLOSE-1                                             *
```

```
*                                                                *
******************************************************************
 90910-CLOSE-1.

     EXEC SQL CLOSE C1 END-EXEC.

 90910-EXIT.
     EXIT.
 EJECT
******************************************************************
*                                                                *
*    90920-CLOSE-2                                               *
*                                                                *
******************************************************************
 90920-CLOSE-2.

     EXEC SQL CLOSE C2 END-EXEC.

 90920-EXIT.
     EXIT.
******************************************************************
*                                                                *
*    90930-CLOSE-3                                               *
*                                                                *
******************************************************************
 90930-CLOSE-3.

     EXEC SQL CLOSE C3 END-EXEC.

 EJECT

******************************************************************
*                                                                *
*    99000-ABNORMAL-TERMINATION                                  *
*                                                                *
******************************************************************
 99000-ABNORMAL-TERMINATION SECTION.
*   All abnormal terminations handled from here.
 9200-DB-ABEND.                                                    0
****************************************************************** 00623000
* THE FOLLOWING ROUTINE PRINTS THE SQLCA STRUCTURE:              * 00624000
*                                                                * 00625000
*  - SQLCODE = SQL RETURN CODE                                   * 00626000
*  - SQLERRM = SQL ERROR MESSAGE                                 * 00627000
*  - SQLERRP = MODULE DETECTING ERROR                            * 00628000
*  - SQLERRD = INTERNAL ERROR VALUES                             * 00629000
*  - SQLWARN = SQL WARNING STRUCTURE                             * 00630000
*                                                                * 00631000
****************************************************************** 00632000
       DISPLAY '*********************************' UPON CONSOLE.  00633000
       DISPLAY '* PROGRAM ERROR ROUTINE ENTERED *' UPON CONSOLE.  00634000
       DISPLAY '* CHECK SYSPRINT FOR ERROR CODES*' UPON CONSOLE.  00635000
       DISPLAY '* CHANGES WILL BE BACKED OUT    *' UPON CONSOLE.  00636000
       DISPLAY '*********************************' UPON CONSOLE.  00637000
       MOVE SQLCODE TO DECODED-SQLCODE.                           00638000
       DISPLAY 'PROGRAM ERROR ROUTINE ENTERED'.                   00639000
       DISPLAY '       *******'.                                  00640000
       DISPLAY 'A PROBLEM HAS BEEN DETECTED IN THE '.             00641000
       DISPLAY STEP-INDICATOR, ' PARAGRAPH.'.                     00642000
       DISPLAY 'THE FOLLOWING ERROR CODES SHOULD AID YOU IN'.     00643000
       DISPLAY 'PROBLEM DETERMINATION OF THE SQL STATEMENT.'.     00644000
       DISPLAY '       *******'.                                  00645000
       DISPLAY 'SQLCODE : ' DECODED-SQLCODE.                      00646000
       DISPLAY 'SQLERRM : ' SQLERRMC.                             00647000
       DISPLAY 'SQLERRP : ' SQLERRP.                              00648000
       PERFORM ERRD VARYING INDX2 FROM 1 BY 1 UNTIL INDX2 = 7.    00649000
```

```
       IF SQLWARN0 NOT EQUAL 'W'                                  00650000
          THEN GO TO 99000-BACKOUT,                               00651000
          ELSE DISPLAY 'SQLWARN0: ' SQLWARN0,                     00652000
               DISPLAY 'SQLWARN1: ' SQLWARN1,                     00653000
               DISPLAY 'SQLWARN2: ' SQLWARN2,                     00654000
               DISPLAY 'SQLWARN3: ' SQLWARN3,                     00655000
               DISPLAY 'SQLWARN4: ' SQLWARN4,                     00656000
               DISPLAY 'SQLWARN5: ' SQLWARN5,                     00657000
               DISPLAY 'SQLWARN6: ' SQLWARN6,                     00658000
               DISPLAY 'SQLWARN7: ' SQLWARN7,                     00659000
               DISPLAY 'SQLWARN8: ' SQLWARN8,                     00660000
               DISPLAY 'SQLWARN9: ' SQLWARN9,                     00661000
               DISPLAY 'SQLWARNA: ' SQLWARNA,                     00662000
               GO TO 99000-BACKOUT.                               00663000

 ERRD.
       MOVE SQLERRD (INDX2) TO DECODED-SQLERRD (INDX2).            0
       DISPLAY 'SQLERRD', INDX2, ': ', DECODED-SQLERRD (INDX2).   00666000
 EXIT-ERROR-ABEND.                                                 0621000
     EXIT.
 eject
 99000-BACKOUT SECTION.                                            00667000
****************************************************************** 00668000
* 'WHENEVER' RESET TO 'CONTINUE' IN THE EVENT THAT THE ROLLBACK  * 00669000
* WORK STATEMENT FAILS TO AVOID LOOP IN ERROR ROUTINE.           * 00670000
****************************************************************** 00671000
                                                                   00672000
     MOVE 'ABEND - BACKING OUT' TO STEP-INDICATOR.                  0674000
                                                                   00675000
     EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.                 00676000
     EXEC SQL ROLLBACK WORK END-EXEC.                              00677000
     STOP RUN.
```

5) Source Code of Module: CPPO0014

```
*
*****************************************************************
*  CPPO0014     Bid Results
*****************************************************************

 IDENTIFICATION DIVISION.
 PROGRAM-ID.  CPPO0014.
 AUTHOR.      Thomas F. Sandish, Cheryl Jones, Ken Everhart,
              Kevin Nester.

*MODIFIED 10/92, STEVE PETER; EMBEDDED STMTS,DYNAMIC CALLS,
*              GOT IT TO WORK WITH OTHER MODULES.
*MODIFIED
*MODIFIED
* PROGRAM FUNCTIONAL REQUIREMENTS
* This program will diplay the bid results to the student using
* screen 1.1.4 View Bid Results. The results to be displayed
* include employer name, winning bid value, student's bid value,
* and the student's scheduling status. The input to this program
* will be the student's social sequrity number from CPPO0000.
* from this key value, the program can retrieve the bid results.
* The values produced for the screen are the outputs of the
* program. The Student ID and Company Name will be transfered
* to program CPPO0015.

 eject
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
     C01 IS TOP-OF-PAGE.

 INPUT-OUTPUT SECTION.
 FILE-CONTROL.


 DATA DIVISION.
 FILE SECTION.

 eject
 WORKING-STORAGE SECTION.
*    Host variables to be used in SQL statements must be          0077000
*    defined in this section.                                    00097000
*
                                                                  STU01960
     EXEC SQL BEGIN DECLARE SECTION END-EXEC.                     STU00010
*    STUDENT COLUMNS FROM STUDENT TABLE.                          STU00020
*    IDENTIFICATION DATA                                          STU00030
 01  SQL-STU-SSN                        PIC X(09).               STU00040
 01  SQL-STU-LAST-NAME                  PIC X(20).               STU00050
 01  SQL-STU-FIRST-NAME                 PIC X(15).               STU00060
 01  SQL-STU-M-INITIAL                  PIC X(01).               STU00070
                                                                  STU00080
*    CPPO DATA                                                    STU00510
 77  SQL-STU-BID-POINTS                 PIC S9(4) COMP.          STU00520
                                                                  STU00530
 01  NI                      PIC S9(4) COMP.

*SQL-BID ENTITY
 01  SQL-BID-SSN                        PIC X(09).
 01  SQL-BID-EMP-NO                     PIC X(05).
 01  SQL-BID-SCH-NO                     PIC X(02).
 01  SQL-BID-POINTS-BID                 PIC S9(4) COMP.
 01  SQL-BID-STATUS                     PIC X(06).
 01  SQL-BID-DATE-OF-BID                PIC X(10).
```

```
* SQL-SCHEDUL ENTITY                                             SCH00030
 01  SQL-SCHEDUL-EMP-NO                 PIC X(5).                SCH00040
 01  SQL-SCHEDUL-SCH-NO                 PIC X(2).                SCH00050
 01  SQL-SCHEDUL-INT-DATE-1             PIC X(10).               SCH00060
 01  SQL-SCHEDUL-INT-DATE-2             PIC X(10).               SCH00060
 01  SQL-SCHEDUL-INT-DATE-3             PIC X(10).               SCH00060
 01  SQL-SCHEDUL-NO-SLOTS             PIC S9(9) COMP.             00350
*01  SQL-SCHEDUL-NO-SLOTS-2             PIC S9(4) COMP.          CH00350
*01  SQL-SCHEDUL-NO-SLOTS-3             PIC S9(4) COMP.          CH00350
 01  SQL-SCHEDUL-SLOTS-TAKEN          PIC S9(9) COMP.
*01  SQL-SCHEDUL-SLOTS-TAKEN-2          PIC S9(4) COMP.
*01  SQL-SCHEDUL-SLOTS-TAKEN-3          PIC S9(4) COMP.
 01  SQL-SCHEDUL-BID-DATE               PIC X(10).               SCH00250
 01  SQL-SCHEDUL-POINTS-USED-TO-WIN     PIC S9(9) COMP.          SCH00270
 01  SQL-SCHEDUL-SIGN-DATE              PIC X(10).               SCH00280
 01  SQL-SCHEDUL-LAST-DATE              PIC X(10).

                                                                  EMP00020
*SQL-EMPLOYER ENTITY                                             EMP00030
 77  SQL-EMP-EMP-NO                     PIC X(05).               EMP00070
 01  SQL-EMP-NAME                       PIC X(40).               EMP00080
                                                                  EMP00120

*SQL-INTERVIE ENTITY
 01  SQL-INTERVIE-EMP-NO                    PIC X(05).
 01  SQL-INTERVIE-SCH-NO                    PIC X(02).
 01  SQL-INTERVIE-DATE                      PIC X(10).
 01  SQL-INTERVIE-TIME                      PIC X(10).
 01  SQL-INTERVIE-SSN                       PIC X(09).
 01  SQL-INTERVIE-SIGN-IN                   PIC X(01).

     EXEC SQL END DECLARE SECTION END-EXEC.                       STU00970
                                                                  STU00980
 eject                                                            STU00990
 01  SQL-STUDENT.                                                STU01000
*    IDENTIFICATION DATA                                          STU01010
     10  STU-STU-SSN                    PIC X(09).               STU01020
     10  STU-LAST-NAME                  PIC X(20).               STU01030
     10  STU-FIRST-NAME                 PIC X(15).               STU01040
     10  STU-M-INITIAL                  PIC X(01).               STU01050
                                                                  STU01480
*    CPPO DATA                                                    STU01490
     10  STU-BID-POINTS                 PIC S9(4).               STU01500
                                                                  STU01510

 01  BID-RECORD.
     10  BID-SSN                        PIC X(09).
     10  BID-EMP-NO                     PIC X(05).
     10  BID-SCH-NO                     PIC X(02).
     10  BID-POINTS-BID                 PIC 9(4).
     10  BID-STATUS                     PIC X(08).
     10  BID-DATE-OF-BID                PIC X(10).

                                                                  SCH00300
 01 SCHEDUL-RECORD.                                               SCH00310
    10  SCHEDUL-EMP-NO                  PIC X(5).                SCH00320
    10  SCHEDUL-SCH-NO                  PIC X(2).                SCH00330
    10  SCHEDUL-INT-DATE-1              PIC X(10).               SCH00340
    10  SCHEDUL-INT-DATE-2              PIC X(10).               SCH00340
    10  SCHEDUL-INT-DATE-3              PIC X(10).               SCH00340
    10  SCHEDUL-NO-SLOTS               PIC 9(4).                  00350
*   10  SCHEDUL-NO-SLOTS-2              PIC 9(4).                SCH00350
*   10  SCHEDUL-NO-SLOTS-3              PIC 9(4).                SCH00350
    10  SCHEDUL-SLOTS-TAKEN            PIC 9(4).
*   10  SCHEDUL-SLOTS-TAKEN-2           PIC 9(4).
```

```
*   10  SCHEDUL-SLOTS-TAKEN-3           PIC 9(4).
    10  SCHEDUL-BID-DATE                PIC X(10).              SCH00530
    10  SCHEDUL-POINTS-USED-TO-WIN      PIC 9(4).               SCH00550
    10  SCHEDUL-SIGN-DATE               PIC X(10).
    10  SCHEDUL-LAST-DATE               PIC X(10).

                                                                EMP00140
 01  EMP-RECORD.                                               EMP00150
     10  EMP-EMP-NO                     PIC X(05).             EMP00190
     10  EMP-NAME                       PIC X(30).             EMP00200


 01  INTERVIE-RECORD.
     10  INTERVIE-EMP-NO                    PIC X(05).
     10  INTERVIE-SCH-NO                    PIC X(02).
     10  INTERVIE-DATE                      PIC X(10).
     10  INTERVIE-TIME                      PIC X(10).
     10  INTERVIE-SSN                       PIC X(09).
     10  INTERVIE-SIGN-IN                   PIC X(01).

 eject                                                          00099000
     EXEC SQL INCLUDE SQLCA END-EXEC.                           00100000
*    Additional variables for abnormal termination.
 01  DECODED-SQLCODE        PIC --------999.                    00107000
                                                                00108000
 01  ARRAY-SQLERRD.                                            00109000
     02 DECODED-SQLERRD     PIC --------999 OCCURS 6 TIMES.     00110000
 01  INDX2                  PIC S9(1) SYNC USAGE IS COMP.       00111000
 01  INDXPIC                PIC ZZZ9.                           00112000


 01  BID-RESULTS-TABLE.
     05  BID-RESULTS-RECORD OCCURS 100 TIMES.
         10  BR-EMP-NO                  PIC X(5).
         10  BR-SCH-NO                  PIC X(2).
         10  BR-EMP-NAME                PIC X(22).
         10  BR-WINNING-BID             PIC XXXX.
         10  BR-POINTS-BID              PIC XXXX.
         10  BR-STATUS-CODE             PIC X(8).
         10  BR-STATUS-MESSAGE          PIC X(39).
         10  BR-BID-DATE                PIC X(10).
         10  BR-INT-DATE-1              PIC X(10).
         10  BR-INT-DATE-2              PIC X(10).
         10  BR-INT-DATE-3              PIC X(10).
         10  BR-CONFIRMATION            PIC X.
         10  BR-WINNER-SIGN-UP          PIC X(10).
         10  BR-LAST-SIGN-UP            PIC X(10).
         10  BR-INTR-DATE               PIC X(10).
         10  BR-INT-TIME                PIC X(10).
         10  BR-NO-SLOTS                PIC 999.
         10  BR-SLOTS-TAKEN             PIC 999.

*   OTHER VARIABLES THAT NEED TO BE DEFINED
 01  I                                  PIC 999.
 01  J                                  PIC 999.
 01  NUMBER-BIDS                        PIC 999 VALUE 0.
 01  PAGE-NUMBER                        PIC 999 VALUE 1.
 01  NUMBER-PAGES                       PIC 999.
 01  LEFTOVER-BIDS                      PIC 999.
 01  NUMERIC-BID                        PIC 9999.
 01  FIRST-REC                          PIC 999 VALUE 1.
 01  LAST-REC                           PIC 999 VALUE 9.
 01  MORE-RECORDS                       PIC X(4) VALUE 'MORE'.
 01  PF7-KEY                            PIC X(11) VALUE
                                                 'PF7:PG BACK'.
 01  PF8-KEY                            PIC X(10) VALUE
                                                 'PF8:PG FWD'.

 01  SCREEN-TABLE.
     05  SCREEN-TABLE-RECORD OCCURS 9 TIMES.
         10  SR-EMP-NO                  PIC X(5).
         10  SR-SCH-NO                  PIC X(2).
         10  SR-EMP-NAME                PIC X(21).
         10  SR-WINNING-BID             PIC XXXXX.
         10  SR-POINTS-BID              PIC XXXXX.
         10  SR-STATUS-MESSAGE          PIC X(39).


* 01 SCRN-FIELDS.
*     05 SCRN-NUMBER     PIC     9(7) COMP.
*     05 PGMID           PIC     X(7).
*     05 SCRNID          PIC     X(8).
*     05 SCRDATE         PIC     X(8).
*     05 SCRTIME         PIC    X(8).
*     05 MORE            PIC     X(4).
*     05 STUID           PIC    X(9).
*     05 PTSREM          PIC     X(5).
*     05 STUNAME         PIC   X(50).
*     05 EMP1            PIC    X(21).
*     05 WBID1           PIC     X(5).
*     05 YBID1           PIC     X(5).
*     05 STAT1           PIC    X(39).
*     05 EMP2            PIC    X(21).
*     05 WBID2           PIC     X(5).
*     05 YBID2           PIC     X(5).
*     05 STAT2           PIC    X(39).
*     05 EMP3            PIC    X(21).
*     05 WBID3           PIC     X(5).
*     05 YBID3           PIC     X(5).
*     05 STAT3           PIC    X(39).
*     05 EMP4            PIC    X(21).
*     05 WBID4           PIC     X(5).
*     05 YBID4           PIC     X(5).
*     05 STAT4           PIC    X(39).
*     05 EMP5            PIC    X(21).
*     05 WBID5           PIC     X(5).
*     05 YBID5           PIC     X(5).
*     05 STAT5           PIC    X(39).
*     05 EMP6            PIC    X(21).
*     05 WBID6           PIC     X(5).
*     05 YBID6           PIC     X(5).
*     05 STAT6           PIC    X(39).
*     05 EMP7            PIC    X(21).
*     05 WBID7           PIC     X(5).
*     05 YBID7           PIC     X(5).
*     05 STAT7           PIC    X(39).
*     05 EMP8            PIC    X(21).
*     05 WBID8           PIC     X(5).
*     05 YBID8           PIC     X(5).
*     05 STAT8           PIC    X(39).
*     05 EMP9            PIC    X(21).
*     05 WBID9           PIC     X(5).
*     05 YBID9           PIC     X(5).
*     05 STAT9           PIC    X(39).
*     05 CMPYNBR         PIC     X(1).
*     05 PFKEY7          PIC    X(11).
*     05 PFKEY8          PIC    X(10).
*     05 SYSMSG          PIC    X(73).
*     05 SCRN-KEY        PIC     X(8).
 ************************************************
```

```
01  SCRN-NUMBER    PIC     9(7) COMP.

01  SCRN-FIELDS.
    05 PGMID           PIC     X(8).
    05 SCRNID          PIC     X(8).
    05 SCRDATE         PIC     X(8).
    05 SCRTIME         PIC     X(8).
    05 MORE            PIC     X(4).
    05 STUID           PIC     X(9).
    05 PTSREM          PIC     X(5).
    05 STUNAME         PIC    X(50).
    05 EMP1            PIC    X(21).
    05 WBID1           PIC     X(5).
    05 YBID1           PIC     X(5).
    05 STAT1           PIC    X(39).
    05 EMP2            PIC    X(21).
    05 WBID2           PIC     X(5).
    05 YBID2           PIC     X(5).
    05 STAT2           PIC    X(39).
    05 EMP3            PIC    X(21).
    05 WBID3           PIC     X(5).
    05 YBID3           PIC     X(5).
    05 STAT3           PIC    X(39).
    05 EMP4            PIC    X(21).
    05 WBID4           PIC     X(5).
    05 YBID4           PIC     X(5).
    05 STAT4           PIC    X(39).
    05 EMP5            PIC    X(21).
    05 WBID5           PIC     X(5).
    05 YBID5           PIC     X(5).
    05 STAT5           PIC    X(39).
    05 EMP6            PIC    X(21).
    05 WBID6           PIC     X(5).
    05 YBID6           PIC     X(5).
    05 STAT6           PIC    X(39).
    05 EMP7            PIC    X(21).
    05 WBID7           PIC     X(5).
    05 YBID7           PIC     X(5).
    05 STAT7           PIC    X(39).
    05 EMP8            PIC    X(21).
    05 WBID8           PIC     X(5).
    05 YBID8           PIC     X(5).
    05 STAT8           PIC    X(39).
    05 EMP9            PIC    X(21).
    05 WBID9           PIC     X(5).
    05 YBID9           PIC     X(5).
    05 STAT9           PIC    X(39).
    05 CMPYNBR         PIC     X(1).
    05 PFKEY7          PIC    X(11).
    05 PFKEY8          PIC    X(10).
    05 SYSMSG          PIC    X(73).
01  SCRN-KEY       PIC     X(8).
    88  PF1                          VALUE 'PF01'.
    88  PF2                          VALUE 'PF02'.
    88  PF3                          VALUE 'PF03'.
    88  PF4                          VALUE 'PF04'.
    88  PF5                          VALUE 'PF05'.
    88  PF6                          VALUE 'PF06'.
    88  PF7                          VALUE 'PF07'.
    88  PF8                          VALUE 'PF08'.
    88  PF9                          VALUE 'PF09'.
    88  PF10                         VALUE 'PF10'.
    88  PF11                         VALUE 'PF11'.
    88  PF12                         VALUE 'PF12'.
    88  RETURN-KEY                   VALUE 'RETURN'.
```

```
*****************************************************
 eject                                                                00101000
*************************************************************** 00102000
*   PROGRAM VARIABLE DECLARTION SECTION                         * 00103000
*************************************************************** 00104000
 01  STEP-INDICATOR          PIC X(30) VALUE 'PREP-SCREEN'.          001050
 01  DB-STATUS-INDICATOR     PIC X(8)  VALUE SPACES.
 01  EDIT-STATUS-INDICATOR   PIC X(8)  VALUE SPACES.
 01  THIS-PROGRAM-ID         PIC X(8)  VALUE 'CPPO0014'.
 01  SCRN-NAME               PIC X(8) VALUE 'SCRN1140'.
 01  SCRN-RETCODE            PIC S9(7) COMP.
 01  SCRN-FIELD-NAME         PIC X(7).
 01  FIELDNAME               PIC X(7).
*01  RETURNED-KEY            PIC X(6).
*    88  PF1                          VALUE 'PF01'.
*    88  PF2                          VALUE 'PF02'.
*    88  PF3                          VALUE 'PF03'.
*    88  PF4                          VALUE 'PF04'.
*    88  PF5                          VALUE 'PF05'.
*    88  PF6                          VALUE 'PF06'.
*    88  PF7                          VALUE 'PF07'.
*    88  PF8                          VALUE 'PF08'.
*    88  PF9                          VALUE 'PF09'.
*    88  PF10                         VALUE 'PF10'.
*    88  PF11                         VALUE 'PF11'.
*    88  PF12                         VALUE 'PF12'.
*    88  RETURN-KEY                   VALUE 'RETURN'.

 01 DATE-TODAY.

     05  DATE-TODAY-YY               PIC XX.
     05  DATE-TODAY-MM               PIC XX.
     05  DATE-TODAY-DD               PIC XX.

 01 TODAY.

     05  FILLER                      PIC XX VALUE '19'.
     05  TODAY-YY                    PIC XX.
     05  FILLER                      PIC X  VALUE '-'.
     05  TODAY-MM                    PIC XX.
     05  FILLER                      PIC X  VALUE '-'.
     05  TODAY-DD                    PIC XX.

 01 SCREEN-DATE.

     05  SCREEN-DATE-MM              PIC XX.
     05  FILLER                      PIC X VALUE '-'.
     05  SCREEN-DATE-DD              PIC XX.
     05  FILLER                      PIC X VALUE '-'.
     05  SCREEN-DATE-YY              PIC XX.

 01 CURRENT-TIME.

     05  HOURS                       PIC XX.
     05  MINUTES                     PIC XX.
     05  FILLER                      PIC XX.

 01 CURTIME.

     05  CURHOURS                    PIC XX.
     05  FILLER                      PIC X VALUE ':'.
     05  CURMINUTES                  PIC XX.


* SYSTEM MESSAGES *
 01 CONFIRMATION-MSG                 PIC X(73)
```

```
           VALUE 'THIS IS NOT A VALID SELECTION'.
01 WRONG-KEY-MSG                          PIC X(73)
           VALUE 'INVALID KEY PRESSED, PLEASE TRY AGAIN'.
01 LINE-MSG                          PIC X(73)
           VALUE 'INVALID LINE NUMBER, PLEASE TRY AGAIN'.

eject

                                                                        00106000
eject                                                                   00113000
01   TERMINAL-MESSAGES.                                                00114000
     02  REC-NOT-FOUND      PIC X(80) VALUE                           00115000
         ' RECORD NOT FOUND - USE A DIFFERENT KEY.'.                    00116000
     02  DUPLICATE-RECORD   PIC X(80) VALUE                           00117000
         ' ATTEMPT TO ADD A DUPLICATE RECORD WAS REJECTED.'.            00118000
eject
LINKAGE SECTION.
*****************************************************************
*    We'll use the same name that CICS uses for data passing.
COPY DFHCOMM.
*****************************************************************
PROCEDURE DIVISION USING DFHCOMMAREA.
    EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
eject
0000-MAINLINE.
    PERFORM 90100-GET-DFHCOMMAREA.
    PERFORM 0500-RETRIEVE-SCREEN.
    PERFORM 1100-HOUSEKEEPING.
    PERFORM 1200-GET-SNAME.
    PERFORM UNTIL STEP-INDICATOR = 'XFER-CONTROL'
       EVALUATE TRUE
           WHEN STEP-INDICATOR = 'PREP-SCREEN'
                PERFORM 1000-PREPARE-SCREEN-OUTPUT
           WHEN STEP-INDICATOR = 'DISP-SCREEN'
                PERFORM 2000-DISPLAY-SCREEN
           WHEN STEP-INDICATOR = 'EVAL-PFKEYS'
*               MOVE SCRN-KEY TO RETURNED-KEY
                PERFORM 3000-EVALUATE-PFKEYS
           WHEN OTHER
*             program logic error if control reaches this point.
              DISPLAY 'STEP INDICATOR NOT SET'
              PERFORM 99000-BACKOUT
       END-EVALUATE
    END-PERFORM.
    PERFORM 4000-TRANSFER-CONTROL.
    GOBACK.
EXIT-MAINLINE.
eject

0500-RETRIEVE-SCREEN.
    CALL 'MLOAD' USING SCRN-NUMBER, SCRN-RETCODE, SCRN-NAME.

    IF SCRN-RETCODE NOT = 0 THEN
     DISPLAY 'ERROR IN RETRIEVING SCREEN'
     PERFORM 99000-BACKOUT
    END-IF.
EXIT-RETRIEVE-SCREEN.

1000-PREPARE-SCREEN-OUTPUT.
*    Performs here to get screen ready.
    PERFORM 1010-LOAD-SCREEN-TABLE.
    PERFORM 1020-MOVE-RECS-TO-SCREEN.
    PERFORM 1030-SET-UP-OTHER-FLDS.
    MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-PREPARE-SCREEN-OUTPUT.
eject
```

```
*  THESE ARE THE PROCEDURES FOR PREPARE SCREEN OUTPUT.  ADD
*  PERFORM STATEMENTS TO CALL THEM

1010-LOAD-SCREEN-TABLE.
    MOVE 1 TO J.
    PERFORM 1011-LOAD-SCREEN-RECS VARYING I FROM FIRST-REC
       BY 1 UNTIL I > LAST-REC.
EXIT-1010-LOAD-SCREEN-TABLE.

1011-LOAD-SCREEN-RECS.
    MOVE BR-EMP-NO(I) TO SR-EMP-NO(J).
    MOVE BR-SCH-NO(I) TO SR-SCH-NO(J).
    MOVE BR-EMP-NAME(I) TO SR-EMP-NAME(J).
    MOVE BR-WINNING-BID(I) TO SR-WINNING-BID(J).
    MOVE BR-POINTS-BID(I) TO SR-POINTS-BID(J).
    MOVE BR-STATUS-MESSAGE(I) TO SR-STATUS-MESSAGE(J).
    ADD 1 TO J.
EXIT-1011-LOAD-SCREEN-RECS.

1020-MOVE-RECS-TO-SCREEN.
    MOVE SR-EMP-NAME(1) TO EMP1.
    MOVE SR-WINNING-BID(1) TO WBID1.
    MOVE SR-POINTS-BID(1) TO YBID1.
    MOVE SR-STATUS-MESSAGE(1) TO STAT1.
    MOVE SR-EMP-NAME(2) TO EMP2.
    MOVE SR-WINNING-BID(2) TO WBID2.
    MOVE SR-POINTS-BID(2) TO YBID2.
    MOVE SR-STATUS-MESSAGE(2) TO STAT2.
    MOVE SR-EMP-NAME(3) TO EMP3.
    MOVE SR-WINNING-BID(3) TO WBID3.
    MOVE SR-POINTS-BID(3) TO YBID3.
    MOVE SR-STATUS-MESSAGE(3) TO STAT3.
    MOVE SR-EMP-NAME(4) TO EMP4.
    MOVE SR-WINNING-BID(4) TO WBID4.
    MOVE SR-POINTS-BID(4) TO YBID4.
    MOVE SR-STATUS-MESSAGE(4) TO STAT4.
    MOVE SR-EMP-NAME(5) TO EMP5.
    MOVE SR-WINNING-BID(5) TO WBID5.
    MOVE SR-POINTS-BID(5) TO YBID5.
    MOVE SR-STATUS-MESSAGE(5) TO STAT5.
    MOVE SR-EMP-NAME(6) TO EMP6.
    MOVE SR-WINNING-BID(6) TO WBID6.
    MOVE SR-POINTS-BID(6) TO YBID6.
    MOVE SR-STATUS-MESSAGE(6) TO STAT6.
    MOVE SR-EMP-NAME(7) TO EMP7.
    MOVE SR-WINNING-BID(7) TO WBID7.
    MOVE SR-POINTS-BID(7) TO YBID7.
    MOVE SR-STATUS-MESSAGE(7) TO STAT7.
    MOVE SR-EMP-NAME(8) TO EMP8.
    MOVE SR-WINNING-BID(8) TO WBID8.
    MOVE SR-POINTS-BID(8) TO YBID8.
    MOVE SR-STATUS-MESSAGE(8) TO STAT8.
    MOVE SR-EMP-NAME(9) TO EMP9.
    MOVE SR-WINNING-BID(9) TO WBID9.
    MOVE SR-POINTS-BID(9) TO YBID9.
    MOVE SR-STATUS-MESSAGE(9) TO STAT9.
EXIT-1020-MOVE-RECS-TO-SCREEN.


1030-SET-UP-OTHER-FLDS.
     IF PAGE-NUMBER < NUMBER-PAGES THEN
        MOVE MORE-RECORDS TO MORE
        MOVE PF8-KEY TO PFKEY8
     ELSE
        MOVE SPACES TO MORE, PFKEY8
```

```
           END-IF.

           IF PAGE-NUMBER > 1 THEN
              MOVE PF7-KEY TO PFKEY7
           ELSE
              MOVE SPACES TO PFKEY7
           END-IF.
           MOVE SPACES TO SYSMSG.
           MOVE 'SCRN1140' TO SCRNID.
           MOVE 'CPPO0014' TO PGMID.
           MOVE DATE-TODAY-YY TO SCREEN-DATE-YY.
           MOVE DATE-TODAY-MM TO SCREEN-DATE-MM.
           MOVE DATE-TODAY-DD TO SCREEN-DATE-DD.
           MOVE SCREEN-DATE TO SCRDATE.
           ACCEPT CURRENT-TIME FROM TIME.
           MOVE HOURS TO CURHOURS.
           MOVE MINUTES TO CURMINUTES.
           MOVE CURTIME TO SCRTIME.
           MOVE '_'     TO CMPYNBR.

       EXIT-1030-SET-UP-OTHER-FLDS.

       1100-HOUSEKEEPING.
      *  THESE ARE THE HOUSEKEEPING PROCEDURES.  PUT A PERFORM
      *  HOUSEKEEPING RIGHT AFTER READING DFHCOMMAREA.

           PERFORM 1109-GET-CURRENT-DATE.
           PERFORM 1110-OPEN-BID-FOR-BROWSE.
           PERFORM 1120-LOAD-BID-RESULTS-TABLE.
           PERFORM 1130-CLOSE-BID-CURSOR.
           PERFORM 1140-DETERMINE-NUMBER-PAGES.
       EXIT-1100-HOUSEKEEPING.

       1109-GET-CURRENT-DATE.
           ACCEPT DATE-TODAY FROM DATE.
      *    MOVE '911212'        TO  DATE-TODAY.
           MOVE DATE-TODAY-YY   TO  TODAY-YY.
           MOVE DATE-TODAY-MM   TO  TODAY-MM.
           MOVE DATE-TODAY-DD   TO  TODAY-DD.

       EXIT-1109-GET-CURRENT-DATE.


       1110-OPEN-BID-FOR-BROWSE.
           MOVE COMM-STUDENT-SSN TO SQL-BID-SSN.
           EXEC SQL DECLARE C_BID CURSOR FOR
                SELECT EMP_NO,SCH_NO,POINTS_BID,
                                     STATUS, DATE_OF_BID
                FROM BID
                WHERE SSN = :SQL-BID-SSN
                ORDER BY DATE_OF_BID DESC
           END-EXEC.

           IF SQLCODE = 0 THEN
              MOVE 'OK' TO DB-STATUS-INDICATOR
           ELSE
                GO TO 99200-DB-ABEND
           END-IF.
           EXEC SQL OPEN C_BID END-EXEC.
       EXIT-1110-OPEN-BID-FOR-BROWSE.

       1120-LOAD-BID-RESULTS-TABLE.
           MOVE SPACES TO BID-RESULTS-TABLE.
           PERFORM 1121-LOAD-BID-RECS
             VARYING I FROM 1 BY 1
               UNTIL SQLCODE = 100.
```

```
           PERFORM 1123-GET-DATA VARYING I FROM 1 BY 1
              UNTIL I > NUMBER-BIDS.
           PERFORM 8000-STATUS-MESSAGE-DETERM
             VARYING I FROM 1 BY 1
               UNTIL I > NUMBER-BIDS.

       EXIT-LOAD-BID-RESULTS-TABLE.

       1121-LOAD-BID-RECS.
           PERFORM 1122-FETCH-BID.
           IF SQLCODE NOT = 100 THEN
              MOVE BID-EMP-NO TO BR-EMP-NO(I)
              MOVE BID-SCH-NO TO BR-SCH-NO(I)
              MOVE BID-POINTS-BID TO NUMERIC-BID
              MOVE NUMERIC-BID TO BR-POINTS-BID(I)
              MOVE BID-STATUS TO BR-STATUS-CODE(I)
              MOVE BID-DATE-OF-BID TO BR-BID-DATE(I)
              ADD 1 TO NUMBER-BIDS
           END-IF.
       EXIT-1121-LOAD-BID-RECS.

       1122-FETCH-BID.
           EXEC SQL FETCH C_BID INTO
                :SQL-BID-EMP-NO,
                :SQL-BID-SCH-NO,
                :SQL-BID-POINTS-BID,
                :SQL-BID-STATUS,
                :SQL-BID-DATE-OF-BID
           END-EXEC.

           IF SQLCODE NOT = 100 THEN
             MOVE SQL-BID-EMP-NO TO BID-EMP-NO
             MOVE SQL-BID-SCH-NO TO BID-SCH-NO
             MOVE SQL-BID-POINTS-BID TO BID-POINTS-BID
             MOVE SQL-BID-STATUS TO BID-STATUS
             MOVE SQL-BID-DATE-OF-BID TO BID-DATE-OF-BID
           END-IF.
       EXIT-1122-FETCH-BID.

       1130-CLOSE-BID-CURSOR.
           EXEC SQL CLOSE C_BID END-EXEC.
       EXIT-1130-CLOSE-BID-CURSOR.

       1140-DETERMINE-NUMBER-PAGES.
           IF NUMBER-BIDS NOT = 0 THEN
              DIVIDE NUMBER-BIDS BY 9
                 GIVING NUMBER-PAGES
                 REMAINDER LEFTOVER-BIDS
              END-DIVIDE
              IF LEFTOVER-BIDS > 0 THEN
                 ADD 1 TO NUMBER-PAGES
              END-IF
           ELSE
              MOVE 1 TO NUMBER-PAGES
           END-IF.
       EXIT-DETERMINE-NUMBER-PAGES.

       1123-GET-DATA.
           MOVE BR-EMP-NO(I) TO SQL-EMP-EMP-NO, SQL-SCHEDUL-EMP-NO,
                                SQL-INTERVIE-EMP-NO.
           MOVE BR-SCH-NO(I) TO SQL-SCHEDUL-SCH-NO,
                                SQL-INTERVIE-SCH-NO.

           EXEC SQL SELECT NAME
                    INTO   :SQL-EMP-NAME
                    FROM   EMPLOYER
```

```
               WHERE  EMP_NO = :SQL-EMP-EMP-NO
           END-EXEC.

           IF SQLCODE = 0 THEN
              MOVE 'OK' TO DB-STATUS-INDICATOR
           ELSE IF SQLCODE = 100
                MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
           ELSE
                GO TO 99200-DB-ABEND
           END-IF.

           MOVE SQL-EMP-NAME TO BR-EMP-NAME(I).

           EXEC SQL SELECT POINTS_USED_TO_WIN, DATE_1_INTERVIEW,
                       DATE_2_INTERVIEW, DATE_3_INTERVIEW,
                       BID_BY_DATE, SCHEDULE_BY_DATE,
                       NUM_TOTAL_SLOTS, NUM_SLOTS_TAKEN
                    INTO   :SQL-SCHEDUL-POINTS-USED-TO-WIN:NI,
                           :SQL-SCHEDUL-INT-DATE-1:NI,
                           :SQL-SCHEDUL-INT-DATE-2:NI,
                           :SQL-SCHEDUL-INT-DATE-3:NI,
                           :SQL-SCHEDUL-SIGN-DATE:NI,
                           :SQL-SCHEDUL-LAST-DATE:NI,
                           :SQL-SCHEDUL-NO-SLOTS:NI,
*                          :SQL-SCHEDUL-NO-SLOTS-2:NI,
*                          :SQL-SCHEDUL-NO-SLOTS-3:NI,
                           :SQL-SCHEDUL-SLOTS-TAKEN:NI
*                          :SQL-SCHEDUL-SLOTS-TAKEN-2:NI,
*                          :SQL-SCHEDUL-SLOTS-TAKEN-3:NI
                    FROM   VISIT
                    WHERE  EMP_NO = :SQL-SCHEDUL-EMP-NO
                    AND    SCH_NO = :SQL-SCHEDUL-SCH-NO
           END-EXEC.
           IF SQLCODE = 0 THEN
              MOVE 'OK' TO DB-STATUS-INDICATOR
           ELSE IF SQLCODE = 100
                MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
           ELSE
                GO TO 99200-DB-ABEND
           END-IF.

           MOVE SQL-SCHEDUL-POINTS-USED-TO-WIN TO
                                        SCHEDUL-POINTS-USED-TO-WIN.
           MOVE SQL-SCHEDUL-INT-DATE-1  TO SCHEDUL-INT-DATE-1.
*          MOVE SQL-SCHEDUL-INT-DATE-2  TO SCHEDUL-INT-DATE-2.
*          MOVE SQL-SCHEDUL-INT-DATE-3  TO SCHEDUL-INT-DATE-3.
           MOVE SQL-SCHEDUL-SIGN-DATE TO SCHEDUL-SIGN-DATE.
           MOVE SQL-SCHEDUL-LAST-DATE TO SCHEDUL-LAST-DATE.
           MOVE SQL-SCHEDUL-NO-SLOTS TO SCHEDUL-NO-SLOTS.
*          MOVE SQL-SCHEDUL-NO-SLOTS-2 TO SCHEDUL-NO-SLOTS-2.
*          MOVE SQL-SCHEDUL-NO-SLOTS-3 TO SCHEDUL-NO-SLOTS-3.
           MOVE SQL-SCHEDUL-SLOTS-TAKEN TO SCHEDUL-SLOTS-TAKEN.
*          MOVE SQL-SCHEDUL-SLOTS-TAKEN-2 TO SCHEDUL-SLOTS-TAKEN-2.
*          MOVE SQL-SCHEDUL-SLOTS-TAKEN-3 TO SCHEDUL-SLOTS-TAKEN-3.
           MOVE SCHEDUL-POINTS-USED-TO-WIN TO BR-WINNING-BID(I).
           MOVE SCHEDUL-INT-DATE-1         TO BR-INT-DATE-1(I).


           MOVE SCHEDUL-SIGN-DATE TO BR-WINNER-SIGN-UP(I).              A
           MOVE SCHEDUL-LAST-DATE TO BR-LAST-SIGN-UP(I).
           MOVE SCHEDUL-NO-SLOTS TO BR-NO-SLOTS(I).
           MOVE SCHEDUL-SLOTS-TAKEN TO BR-SLOTS-TAKEN(I).
*          ADD SCHEDUL-NO-SLOTS SCHEDUL-NO-SLOTS-2
*              SCHEDUL-NO-SLOTS-3 TO BR-NO-SLOTS(I).
*          ADD SCHEDUL-SLOTS-TAKEN SCHEDUL-SLOTS-TAKEN-2
*              SCHEDUL-SLOTS-TAKEN-3 TO BR-SLOTS-TAKEN(I).
```

```
           EXEC SQL SELECT INT_DATE, INT_TIME
                    INTO :SQL-INTERVIE-DATE, :SQL-INTERVIE-TIME
                    FROM INTERVIE
                    WHERE SSN = :SQL-INTERVIE-SSN
                    AND EMP_NO = :SQL-INTERVIE-EMP-NO
                    AND SCH_NO = :SQL-INTERVIE-SCH-NO
           END-EXEC.

           IF SQLCODE NOT = 100 THEN
              MOVE SQL-INTERVIE-DATE TO BR-INTR-DATE(I)
              MOVE SQL-INTERVIE-TIME TO BR-INT-TIME(I)
           ELSE
              MOVE SPACES TO BR-INTR-DATE(I)
           END-IF.
       EXIT-1123-GET-DATA.

       1200-GET-SNAME.
           EXEC SQL SELECT FIRST_NAME, M_INITIAL, LAST_NAME,
                           BID_POINTS_AVAIL
                INTO   :SQL-STU-FIRST-NAME, :SQL-STU-M-INITIAL,
                       :SQL-STU-LAST-NAME, :SQL-STU-BID-POINTS
                FROM   STUDENT
                WHERE  SSN = :SQL-STU-SSN
           END-EXEC.
           IF SQLCODE = 0 THEN
              MOVE 'OK' TO DB-STATUS-INDICATOR
           ELSE IF SQLCODE = 100
                MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
           ELSE
                GO TO 99200-DB-ABEND
           END-IF.

           STRING
              SQL-STU-FIRST-NAME DELIMITED BY SPACE
              ' ' DELIMITED BY SIZE
              SQL-STU-M-INITIAL DELIMITED BY SPACE
              ' ' DELIMITED BY SIZE
              SQL-STU-LAST-NAME DELIMITED BY SPACE
              INTO STUNAME
           END-STRING.

           MOVE SQL-STU-BID-POINTS TO STU-BID-POINTS.
           MOVE STU-BID-POINTS TO PTSPREM.
       EXIT-1200-GET-SNAME.

       8000-STATUS-MESSAGE-DETERM.

           EVALUATE TRUE
              WHEN BR-STATUS-CODE(I) = 'LOST'
                 PERFORM 8100-VALIDATE-LOSING-BID
              WHEN (BR-STATUS-CODE(I) = 'WON' OR
                    BR-STATUS-CODE(I) = 'CLOSED')
                 PERFORM 8200-VALIDATE-WINNING-BID
              WHEN BR-STATUS-CODE(I) = 'INTSETUP'
                 PERFORM 8400-VALIDATE-SCHDLD-INT
              WHEN BR-STATUS-CODE(I) = 'CANCLDINT'
                 PERFORM 8500-CNCLD-INT-MSG
              WHEN BR-STATUS-CODE(I) = 'EMPCANCL'
                 PERFORM 8600-EMP-CNCLD-INT-MSG
              WHEN BR-STATUS-CODE(I) = 'ABSENT'
                 PERFORM 8700-STU-ABSENT-MSG
              WHEN OTHER
                 PERFORM 99000-BACKOUT
           END-EVALUATE.
```

```
EXIT-STATUS-MESSAGE-DETERM.

8100-VALIDATE-LOSING-BID.
    IF (TODAY >= BR-INT-DATE-1(I)) THEN
       MOVE 'Losing bid; all interviews completed.' TO
          BR-STATUS-MESSAGE(I)
       MOVE 'N' TO BR-CONFIRMATION(I)
    ELSE IF (TODAY > BR-WINNER-SIGN-UP(I)) AND (BR-NO-SLOTS(I) >
            BR-SLOTS-TAKEN(I)) AND (TODAY <= BR-LAST-SIGN-UP(I))
          THEN
       STRING
           'REOPENED UNTIL ' DELIMITED BY SIZE
           BR-LAST-SIGN-UP(I) DELIMITED BY SPACE
           ' .' DELIMITED BY SIZE
           INTO BR-STATUS-MESSAGE(I)
       END-STRING
       MOVE 'Y' TO BR-CONFIRMATION(I)
    ELSE
       MOVE 'Losing bid; possible to schedule later.' TO
           BR-STATUS-MESSAGE(I)
       MOVE 'N' TO BR-CONFIRMATION(I)
    END-IF.
EXIT-8100-VALIDATE-LOSING-BID.

8200-VALIDATE-WINNING-BID.

    IF (TODAY >= BR-INT-DATE-1(I))  THEN
       MOVE 'Interview never scheduled.' TO
           BR-STATUS-MESSAGE(I)
       MOVE 'N' TO BR-CONFIRMATION(I)
    ELSE IF (TODAY > BR-WINNER-SIGN-UP(I)) AND (BR-NO-SLOTS(I) >
            BR-SLOTS-TAKEN(I)) AND (TODAY <= BR-LAST-SIGN-UP(I))
         THEN
       STRING
           'REOPENED UNTIL ' DELIMITED BY SIZE
           BR-LAST-SIGN-UP(I) DELIMITED BY SPACE
           ' .' DELIMITED BY SIZE
           INTO BR-STATUS-MESSAGE(I)
       END-STRING
       MOVE 'Y' TO BR-CONFIRMATION(I)
    ELSE IF (TODAY > BR-WINNER-SIGN-UP(I)) AND (BR-NO-SLOTS(I) =
            BR-SLOTS-TAKEN(I)) THEN
       MOVE 'Interview never scheduled.' TO
           BR-STATUS-MESSAGE(I)
       MOVE 'N' TO BR-CONFIRMATION(I)
    ELSE
       PERFORM 8300-SCHD-INT-MSG
    END-IF.
EXIT-8200-VALIDATE-WINNING-BID.

8300-SCHD-INT-MSG.
    STRING
        'SCHEDULE AN INTERVIEW BY ' DELIMITED BY SIZE
*       BR-WINNER-SIGN-UP(I) DELIMITED BY SPACE
        BR-LAST-SIGN-UP(I) DELIMITED BY SPACE
        ' .' DELIMITED BY SIZE
        INTO BR-STATUS-MESSAGE(I)
    END-STRING.
    MOVE 'Y' TO BR-CONFIRMATION(I).
EXIT-8300-SCHD-INT-MSG.

8400-VALIDATE-SCHDLD-INT.
    IF TODAY > BR-INT-DATE-1(I) THEN
       MOVE 'Interview was held.' TO
          BR-STATUS-MESSAGE(I)
       MOVE 'N' TO BR-CONFIRMATION(I)
```

```
    ELSE
        STRING
            'INTERVIEW ON ' DELIMITED BY SIZE
            BR-INTR-DATE(I) DELIMITED BY SPACE
            ' AT ' DELIMITED BY SIZE
            BR-INT-TIME(I) DELIMITED BY SPACE
            ' .' DELIMITED BY SIZE
            INTO BR-STATUS-MESSAGE(I)
        END-STRING
        MOVE 'Y' TO BR-CONFIRMATION(I)
    END-IF.
EXIT-8400-VALIDATE-SCHDLD-INT.

8500-CNCLD-INT-MSG.
    MOVE 'You cancelled your interview.'  TO
        BR-STATUS-MESSAGE(I).
    MOVE 'N' TO BR-CONFIRMATION(I).
EXIT-8500-CNCLD-INT-MSG.

8600-EMP-CNCLD-INT-MSG.
    MOVE 'The employer cancelled interviews.' TO
        BR-STATUS-MESSAGE(I).
    MOVE 'N' TO BR-CONFIRMATION(I).
EXIT-8600-EMP-CNCLD-INT-MSG.

8700-STU-ABSENT-MSG.
    MOVE 'You never showed up for your interview.' TO
        BR-STATUS-MESSAGE(I).
    MOVE 'N' TO BR-CONFIRMATION(I).
EXIT-8700-STU-ABSENT-MSG.


2000-DISPLAY-SCREEN.
*   CALL 'MPCUR' USING SCRN-NUMBER SCRN-RETCODE FIELDNAME
    CALL 'SCRN1140' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.
    MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.
EXIT-DISPLAY-SCREEN.
eject
3000-EVALUATE-PFKEYS.
    EVALUATE TRUE
       WHEN PF1
*           CALL 'CPPOHELP' USING SCRN-NAME
            MOVE 'DISP-SCREEN' TO STEP-INDICATOR
       WHEN PF3
            MOVE 'CICS     ' TO NEXT-PROGRAM-ID
            MOVE THIS-PROGRAM-ID     TO PREV-PROGRAM-ID
            MOVE 'XFER-CONTROL' TO STEP-INDICATOR
       WHEN PF4
            IF (CMPYNBR >= '1') AND (CMPYNBR <= '9') THEN
               MOVE CMPYNBR TO I
               IF BR-CONFIRMATION(I) = 'N' THEN
                  MOVE CONFIRMATION-MSG TO SYSMSG
                  MOVE 'DISP-SCREEN' TO STEP-INDICATOR
               ELSE
                  MOVE 'CPP00015'      TO NEXT-PROGRAM-ID
                  MOVE THIS-PROGRAM-ID     TO PREV-PROGRAM-ID
                  MOVE SR-EMP-NO(I)    TO COMM-EMP-NUM
                  MOVE SR-SCH-NO(I)  TO COMM-SCH-NUM
                  MOVE 'XFER-CONTROL'  TO STEP-INDICATOR
               END-IF
            ELSE
               MOVE LINE-MSG TO SYSMSG
               MOVE 'DISP-SCREEN' TO STEP-INDICATOR
            END-IF
       WHEN PF7
           IF PAGE-NUMBER = 1 THEN
```

```
                MOVE 1 TO FIRST-REC
                MOVE 9 TO LAST-REC
             ELSE
                SUBTRACT 9 FROM FIRST-REC, LAST-REC
                SUBTRACT 1 FROM PAGE-NUMBER
             END-IF

             MOVE 'PREP-SCREEN' TO STEP-INDICATOR
          WHEN PF8
             IF PAGE-NUMBER NOT = NUMBER-PAGES THEN
                ADD 9 TO FIRST-REC, LAST-REC
                ADD 1 TO PAGE-NUMBER
             END-IF
             MOVE 'PREP-SCREEN' TO STEP-INDICATOR
          WHEN OTHER
             MOVE WRONG-KEY-MSG TO SYSMSG
             MOVE 'DISP-SCREEN' TO STEP-INDICATOR
       END-EVALUATE.
 EXIT-EVALUATE-PFKEYS.
 eject
 4000-TRANSFER-CONTROL.
      CALL 'MPURGE' USING SCRN-NUMBER, SCRN-RETCODE.

      IF SCRN-RETCODE NOT = 0 THEN
       DISPLAY 'ERROR IN EXITING SCREEN'
       PERFORM 99000-BACKOUT
      END-IF.

      MOVE 0 TO RETURN-CODE.
 EXIT-TRANSFER-CONTROL.
 eject

*
* Replace 90100-GET-DFHCOMMAREA SECTION. with this paragraph.
*
 90100-GET-DFHCOMMAREA.
*    MOVE '222222222' TO COMM-STUDENT-SSN.

      MOVE COMM-STUDENT-SSN TO STUID, SQL-STU-SSN,
             SQL-INTERVIE-SSN.
 EXIT-90100-GET-DFHCOMMAREA.

 eject

 99000-ABNORMAL-TERMINATION SECTION.
*   All abnormal terminations handled from here.
 99200-DB-ABEND.                                                    0
*****************************************************************  00623000
* THE FOLLOWING ROUTINE PRINTS THE SQLCA STRUCTURE:              *  00624000
*                                                                *  00625000
*  - SQLCODE = SQL RETURN CODE                                   *  00626000
*  - SQLERRM = SQL ERROR MESSAGE                                 *  00627000
*  - SQLERRP = MODULE DETECTING ERROR                            *  00628000
*  - SQLERRD = INTERNAL ERROR VALUES                             *  00629000
*  - SQLWARN = SQL WARNING STRUCTURE                             *  00630000
*                                                                *  00631000
*****************************************************************  00632000
       DISPLAY '*********************************' UPON CONSOLE.   00633000
       DISPLAY '* PROGRAM ERROR ROUTINE ENTERED *' UPON CONSOLE.   00634000
       DISPLAY '* CHECK SYSPRINT FOR ERROR CODES*' UPON CONSOLE.   00635000
       DISPLAY '* CHANGES WILL BE BACKED OUT    *' UPON CONSOLE.   00636000
       DISPLAY '*********************************' UPON CONSOLE.   00637000
       MOVE SQLCODE TO DECODED-SQLCODE.                            00638000
       DISPLAY 'PROGRAM ERROR ROUTINE ENTERED'.                    00639000
       DISPLAY '       *******'.                                   00640000
       DISPLAY 'A PROBLEM HAS BEEN DETECTED IN THE '.              00641000
```

```
       DISPLAY STEP-INDICATOR, ' PARAGRAPH.'.                      00642000
       DISPLAY 'THE FOLLOWING ERROR CODES SHOULD AID YOU IN'.      00643000
       DISPLAY 'PROBLEM DETERMINATION OF THE SQL STATEMENT.'.      00644000
       DISPLAY '       *******'.                                   00645000
       DISPLAY 'SQLCODE : ' DECODED-SQLCODE.                       00646000
       DISPLAY 'SQLERRM : ' SQLERRMC.                              00647000
       DISPLAY 'SQLERRP : ' SQLERRP.                               00648000
       PERFORM ERRD VARYING INDX2 FROM 1 BY 1 UNTIL INDX2 = 7.     00649000
       IF SQLWARN0 NOT EQUAL 'W'                                   00650000
           THEN GO TO 99000-BACKOUT,                               00651000
           ELSE DISPLAY 'SQLWARN0: ' SQLWARN0,                     00652000
                DISPLAY 'SQLWARN1: ' SQLWARN1,                     00653000
                DISPLAY 'SQLWARN2: ' SQLWARN2,                     00654000
                DISPLAY 'SQLWARN3: ' SQLWARN3,                     00655000
                DISPLAY 'SQLWARN4: ' SQLWARN4,                     00656000
                DISPLAY 'SQLWARN5: ' SQLWARN5,                     00657000
                DISPLAY 'SQLWARN6: ' SQLWARN6,                     00658000
                DISPLAY 'SQLWARN7: ' SQLWARN7,                     00659000
                DISPLAY 'SQLWARN8: ' SQLWARN8,                     00660000
                DISPLAY 'SQLWARN9: ' SQLWARN9,                     00661000
                DISPLAY 'SQLWARNA: ' SQLWARNA,                     00662000
                GO TO 99000-BACKOUT.                               00663000

 ERRD.
       MOVE SQLERRD (INDX2) TO DECODED-SQLERRD (INDX2).             0
       DISPLAY 'SQLERRD', INDX2, ': ', DECODED-SQLERRD (INDX2).    00666000
 EXIT-ERROR-ABEND.                                                  0621000
 eject
 99000-BACKOUT SECTION.                                             00667000
*****************************************************************  00668000
* 'WHENEVER' RESET TO 'CONTINUE' IN THE EVENT THAT THE ROLLBACK *  00669000
* WORK STATEMENT FAILS TO AVOID LOOP IN ERROR ROUTINE.          *  00670000
*****************************************************************  00671000
                                                                    00672000
      MOVE 'ABEND - BACKING OUT' TO STEP-INDICATOR.                  0674000
                                                                    00675000
      EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.                 00676000
      EXEC SQL ROLLBACK WORK END-EXEC.                              00677000
      GOBACK.
```

6) Source Code of Module: CPPO0015

```
*
*****************************************************************
*  CPPO0015  Interview Scheduling Program
*****************************************************************

 IDENTIFICATION DIVISION.
 PROGRAM-ID.  CPPO0015.
 AUTHOR.      JR2GSANU:J Rawlings.

*MODIFY LIST:
*MODIFIED 10/92, STEVE PETER; EMBEDDED STMTS,DYNAMIC CALLS,
*                GOT IT TO WORK WITH OTHER MODULES.
*MODIFIED
*MODIFIED
* PROGRAM FUNCTIONAL REQUIREMENTS
* This program uses the INTERVIE and EMPLOYER tables.
* Variables passed to this program are SSN,EMP_NO and SCH_NO
* and they are already in a "validated" state.  Control is
* not passed to this program if the student is invalid or
* unable to meet requirements for the interview scheduling.
* This program only assists in selection/de-selection of
* interview DATE/TIME based on EMP_NO:SCH_NO.
* Data sent back to the calling program (CPPO0014) would be
* SSN, EMP_NO, SCH_NO, INT_DATE, INT_TIME.  A subroutine exists
* for sending and updating bid STATUS codes.  These codes assist
* in informing as to whether the student is signed up 'N' or  as
* to whether he/she has cancelled 'X' his/her interview.
*
* Group members assigned to this 'Backbone' program:
*               MC37SANU
*               KA0HSANU
*               JG11SANU
*
*          Last revision: 25 NOV 1991
*          Total Hours:   36
*
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
     C01 IS TOP-OF-PAGE.

 INPUT-OUTPUT SECTION.
 FILE-CONTROL.


 DATA DIVISION.
 FILE SECTION.
 eject
 WORKING-STORAGE SECTION.                                          0077000

     EXEC SQL BEGIN DECLARE SECTION END-EXEC.

*SQL-EMPLOYER ENTITY                                               EMP00030
 01  SQL-EMP-ADDRESS                  PIC X(20).                   EMP00040
 01  SQL-EMP-CITY                     PIC X(20).                   EMP00050
 01  SQL-EMP-CONTACT                  PIC X(20).                   EMP00060
 77  SQL-EMP-EMP-NO                   PIC X(05).                   EMP00070
 01  SQL-EMP-NAME                     PIC X(30).                   EMP00080
 77  NUL-IND-EMPNAME                  PIC S9(4) COMP.
 01  SQL-EMP-PHONE                    PIC X(10).                   EMP00090
 01  SQL-EMP-STATE                    PIC X(02).                   EMP00100
 01  SQL-EMP-ZIP                      PIC X(05).                   EMP00110
                                                                   00099000
*SQL-INTERVIE ENTITY
 01  SQL-INTERVIE-EMP-NO                   PIC X(05).
 01  SQL-INTERVIE-SCH-NO                   PIC X(02).
```

```
 01  SQL-INTERVIE-DATE                     PIC X(10).
 01  SQL-INTERVIE-TIME                     PIC X(10).
 01  SQL-INTERVIE-ROOM                     PIC X(03).
 01  SQL-INTERVIE-INTNO                    PIC X(02).
 77  NUL-ROOM-NO                           PIC S9(4) COMP.
 01  SQL-INTERVIE-SSN                 PIC X(09)  VALUE SPACES.
 77  NUL-IND-SSN                           PIC S9(4) COMP.
 01  SQL-INTERVIE-LITERATURE-REC           PIC X(01).
 01  SQL-INTERVIE-SIGN-IN                  PIC X(01).

* SQL-SCHEDUL ENTITY
 01  SQL-SCHEDUL-EMP-NO               PIC X(5).
 01  SQL-SCHEDUL-SCH-NO               PIC X(2).
 01  SQL-SCHEDUL-SLOTS-TAKEN          PIC S9(2) COMP.


*SQL-BID ENTITY
 01  SQL-BID-SSN                      PIC X(09).
 01  SQL-BID-EMP-NO                   PIC X(05).
 01  SQL-BID-SCH-NO                   PIC X(02).
 77  SQL-BID-POINTS-BID               PIC S9(4) COMP.
 01  SQL-BID-STATUS                   PIC X(01).
 01  SQL-BID-DATE-OF-BID              PIC X(10).


 eject

     EXEC SQL END DECLARE SECTION END-EXEC.

     EXEC SQL INCLUDE SQLCA END-EXEC.                              00100000
*    Additional variables for abnormal termination.
 01  DECODED-SQLCODE        PIC --------999.                       00107000
                                                                   00108000
 01  ARRAY-SQLERRD.                                               00110000
     02 DECODED-SQLERRD      PIC --------999 OCCURS 6 TIMES.      00111000
 01  INDX2                  PIC S9(1) SYNC USAGE IS COMP.         00111200
 01  INDXPIC                PIC ZZZ9.                             00112000
 eject                                                             00101000
***************************************************************** 00102000
*    PROGRAM VARIABLE DECLARATION SECTION                       * 00103000
***************************************************************** 00104000
 01  STEP-INDICATOR         PIC X(30) VALUE 'PPEP-SCREEN'.          001050
 01  DB-STATUS-INDICATOR    PIC X(8)  VALUE SPACES.
 01  EDIT-STATUS-INDICATOR  PIC X(8)  VALUE SPACES.
 01  THIS-PROGRAM-ID        PIC X(8)  VALUE 'CPPO0015'.
 01  SCRN-NAME              PIC X(8) VALUE 'SCRN1141'.
 01  SCRN-RETCODE           PIC S9(7) COMP.
 01  FIELDNAME              PIC X(7).
 01  PROT                   PIC X(7).
 01  BRIGHT                 PIC X(7).


 eject
 01  EMP-RECORD.                                                  EMP00150
     10  EMP-ADDRESS                  PIC X(20).                  EMP00160
     10  EMP-CITY                     PIC X(20).                  EMP00170
     10  EMP-CONTACT                  PIC X(20).                  EMP00180
     10  EMP-EMP-NO                   PIC X(05).                  EMP00190
     10  EMP-NAME                     PIC X(30).                  EMP00200
     10  EMP-PHONE                    PIC X(10).                  EMP00210
     10  EMP-STATE                    PIC X(02).                  EMP00220
     10  EMP-ZIP                      PIC X(05).                  EMP00230


 01  INTERVIE-RECORD.
     10  INTERVIE-EMP-NO                PIC X(05).
```

```
    10  INTERVIE-SCH-NO                     PIC X(02).
    10  INTERVIE-DATE                       PIC X(10).
    10  INTERVIE-TIME                       PIC X(10).
    10  INTERVIE-ROOM                       PIC X(03).
    10  INTERVIE-SSN                        PIC X(09).
    10  INTERVIE-LITERATURE-REC             PIC X(01).
    10  INTERVIE-SIGN-IN                    PIC X(01).


01 SCHEDUL-RECORD.
   10  SCHEDUL-EMP-NO                  PIC X(5).
   10  SCHEDUL-SCH-NO                  PIC X(2).
   10  SCHEDUL-SLOTS-TAKEN             PIC 99.


01  BID-RECORD.
    10  BID-SSN                         PIC X(09).
    10  BID-EMP-NO                      PIC X(05).
    10  BID-SCH-NO                      PIC X(02).
    10  BID-POINTS-BID                  PIC 9(4).
    10  BID-STATUS                      PIC X(01).
    10  BID-DATE-OF-BID                 PIC X(10).


eject
01  SQL-STUDENT.                                                    STU00900
*   IDENTIFICATION DATA                                             STU00910
    10  STU-SQL-STU-SSN                 PIC X(09).                 STU00920


01  DATE-TIME-DEF.                                                  STU01750
    05  WS-DATE                         PIC X(10).
    05  WS-DATE-X     REDEFINES  WS-DATE.
        10  CENTURY                     PIC XX.
        10  D-YEAR                      PIC XX.
        10  VIRGULE-1                   PIC X.
        10  D-MTH                       PIC XX.
        10  VIRGULE-2                   PIC X.
        10  D-DAY                       PIC XX.
    05  WS-TIME                         PIC X(10).
    05  WS-TIME-X     REDEFINES  WS-TIME.
        10  T-HOUR                      PIC XX.
        10  VIRGULE-3                   PIC X.
        10  T-MIN                       PIC XX.
        10  VIRGULE-4                   PIC X(5).
    05  DDD-X                           PIC XX.
    05  DDD            REDEFINES DDD-X  PIC 99.

eject
01  DATE-TIME-STDS.
    05  D-STD.
        10  STD-M                       PIC XX.
        10  FILLER                      PIC X   VALUE '/'.
        10  STD-D                       PIC XX.
        10  FILLER                      PIC X   VALUE '/'.
        10  STD-Y                       PIC XX.
    05  T-STD.
        10  STD-HR                      PIC XX.
        10  FILLER                      PIC X   VALUE ':'.
        10  STD-MN                      PIC XX.
    05  SCR-TIME.
        10  HR                          PIC XX.
        10  FILLER                      PIC X VALUE ':'.
        10  MN                          PIC XX.
```

```
        10  FILLER                      PIC X VALUE ':'.
        10  SC                          PIC XX.

01  JUL-DATE                            PIC 999.
01  C-JUL                               PIC 999.
01  L-JUL                               PIC 999.
01  DAY-DIFF                            PIC 999.

01  MILITARY-DATE                       PIC 9(6).
01  DATE-PIECES       REDEFINES MILITARY-DATE.
    05  YR                              PIC XX.
    05  MO                              PIC XX.
    05  DY                              PIC XX.


01  CURR-TIME                           PIC 9(8).
01  CURR-TIME-X        REDEFINES CURR-TIME.
    05  HH                              PIC XX.
    05  MM                              PIC XX.
    05  SS                              PIC XX.
    05  TT                              PIC XX.

01  SLOT-ST-N             PIC X(9)   VALUE 'AVAILABLE'.
01  SLOT-ST-Y             PIC X(9)   VALUE 'TAKEN    '.
01  SLOT-ST-P             PIC X(9)   VALUE '**YOURS**'.


01  PFKEY-STATUS.
    05  CANCEL-OK                PIC X.
    05  PGF-OK                   PIC X.
    05  PGB-OK                   PIC X.
    05  PRC-OK                   PIC X.
    05  OK-DATE                  PIC X.

01  CHECK-SLOT                   PIC 99.
01  CONT-LOGIC                   PIC X.

01  ACTING-PASSED-VARS.
    05  STUDENT-SSN              PIC X(9).
 eject

*************************************************************
*********  TABLE AREA
***********************************


01  INT-SCHED-ARRAY.
    05  TOTAL-DATES OCCURS 100 TIMES.
        10  TABLE-NUM                   PIC 9.
        10  TABLE-NDX                   PIC 999.
        10  SLOT-STATUS                 PIC X.
        10  I-SLOT-NUM                  PIC XX.
        10  I-DATE                      PIC X(8).
        10  I-TIME                      PIC X(5).
        10  I-ROOM                      PIC XXX.
        10  STUD-SSN                    PIC X(9).
        10  I-INTNO                     PIC X(2).

01  SUB1                                PIC 999.

01  SLOTT.
    05 SLOT-X                          PIC 99.
    05  SLOT-N  REDEFINES SLOT-X       PIC XX.

01  SCR-FLD-NAMES.
    05  FLD-NAM    OCCURS 13 TIMES        PIC X(75).
```

```
eject

01  STUD-INT-LOGIC.
    05  EXIST-AT-ALL                   PIC X.
    05  EXIST-PAGE                     PIC 9.
    05  EXIST-NDX                      PIC 999 VALUE ZERO.
    05  EXIST-SLOT                     PIC 99.
    05  EXIST-DATE                     PIC X(8).
    05  EXIST-TIME                     PIC X(5).

01  PAG-LOGIC.
    05  PAGE-NUM                       PIC 9.
    05  MAX-PAGE                       PIC 9.
    05  MAX-SUB1                       PIC 999.

01  TOP-PAGE-NDX.
    05  STRING-2                       PIC X(21) VALUE
        '001014027040053066079'.
    05  TOP-OF-PG-NDX REDEFINES STRING-2 OCCURS 7 TIMES PIC 999.

01  NDX-S.
    05  NDX-PRIOR                      PIC 999.
    05  NDX-START                      PIC 999.
    05  NDX-END                        PIC 999.
    05  SR                             PIC 99.

                                                                    00106000
eject                                                               00113000
01  TERMINAL-MESSAGES.                                             00114000
    02  REC-NOT-FOUND      PIC X(80) VALUE                         00115000
      ' RECORD NOT FOUND - USE A DIFFERENT KEY.'.                   00116000
    02  DUPLICATE-RECORD   PIC X(80) VALUE                         00117000
      ' ATTEMPT TO ADD A DUPLICATE RECORD WAS REJECTED.'.           00118000

*****************************************************************
****
****    screen and program linkages
************************************


01  SCRN-NUMBER    PIC     9(7) COMP.

01  SCRN-FIELDS.
    05 PGMID           PIC     X(8) VALUE 'CPPO0015'.
    05 SCRNID          PIC     X(8) VALUE 'SCRN1141'.
    05 SCRDATE         PIC     X(8).
    05 SCRTIME         PIC     X(8).
    05 STUDID          PIC     X(9).
    05 EMPNUM          PIC     X(5).
    05 SCHDNUM         PIC     X(2).
    05 EMPNAME         PIC    X(30).
    05 MORSLOT         PIC     X(4).
    05 ROW01           PIC    X(75).
    05 ROW02           PIC    X(75).
    05 ROW03           PIC    X(75).
    05 ROW04           PIC    X(75).
    05 ROW05           PIC    X(75).
    05 ROW06           PIC    X(75).
    05 ROW07           PIC    X(75).
    05 ROW08           PIC    X(75).
    05 ROW09           PIC    X(75).
    05 ROW10           PIC    X(75).
    05 ROW11           PIC    X(75).
    05 ROW12           PIC    X(75).
```

```
    05 ROW13           PIC    X(75).
    05 ACTMSG          PIC    X(56).
    05 SLOTSEL         PIC     X(2).
    05 PFONE           PIC     X(8).
    05 PFTHREE         PIC     X(8).
    05 PFFOUR          PIC    X(25).
    05 PFSEVEN         PIC    X(10).
    05 PFEIGHT         PIC    X(10).
    05 SYSMSG          PIC    X(78).
01  SCRN-KEY   PIC    X(8).
        88 PF1                VALUE 'PF01'.
        88 PF2                VALUE 'PF02'.
        88 PF3                VALUE 'PF03'.
        88 PF4                VALUE 'PF04'.
        88 PF5                VALUE 'PF05'.
        88 PF6                VALUE 'PF06'.
        88 PF7                VALUE 'PF07'.
        88 PF8                VALUE 'PF08'.
        88 PF9                VALUE 'PF09'.
        88 PF10               VALUE 'PF10'.
        88 PF11               VALUE 'PF11'.
        88 PF12               VALUE 'PF12'.
        88 RETURN-KEY         VALUE 'RETURN'.
eject
LINKAGE SECTION.

    COPY DFHCOMM.
eject


PROCEDURE DIVISION USING DFHCOMMAREA.
    EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
    PERFORM 90100-GET-DFHCOMMAREA.
    PERFORM 90010-GET-SELECTED-EMP.
    IF DB-STATUS-INDICATOR = 'NOT FOUND'
       DISPLAY REC-NOT-FOUND
       PERFORM 99000-BACKOUT
    END-IF.
eject
0000-MAINLINE SECTION.
    CALL 'MLOAD'  USING SCRN-NUMBER SCRN-RETCODE SCRN-NAME.
    IF SCRN-RETCODE NOT = 0 THEN
        DISPLAY 'SCREEN PROGRAM SCRN1141 NOT FOUND!'
        PERFORM 99000-BACKOUT.
    PERFORM 77000-TABLE-FILL.
    PERFORM UNTIL STEP-INDICATOR = 'XFER-CONTROL'
       EVALUATE TRUE
           WHEN STEP-INDICATOR = 'PREP-SCREEN'
                PERFORM 1000-PREPARE-SCREEN-OUTPUT
           WHEN STEP-INDICATOR = 'DISP-SCREEN'
                PERFORM 2000-DISPLAY-SCREEN
           WHEN STEP-INDICATOR = 'EVAL-PFKEYS'
                PERFORM 3000-EVALUATE-PFKEYS
           WHEN OTHER
              DISPLAY 'STEP INDICATOR NOT SET'
              PERFORM 99000-BACKOUT
       END-EVALUATE
    END-PERFORM.
    PERFORM 4000-TRANSFER-CONTROL.
    CALL 'MPURGE' USING SCRN-NUMBER SCRN-RETCODE.
    GOBACK.
EXIT-MAINLINE.

*    This area above contains MAIN program logic and
*    initialization of variables/screens/tabled datum
```

```
eject

PREPARE-SCREEN SECTION.

1000-PREPARE-SCREEN-OUTPUT.
    IF EXIST-AT-ALL = 'Y' THEN
    PERFORM 5100-FILL-SCREEN-EXIST
    ELSE
    PERFORM 5500-FILL-SCREEN-NOTEXIST
    END-IF.
    PERFORM 7100-SET-SCR-ATTS-BY-LINE.
    PERFORM 7500-FINISH-FILL.
    MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-PREPARE-SCREEN-OUTPUT.

*    Initial screen preperation; if student is already signed up
*    then the page (from table) that contains 'that' interview
*    data is the FIRST displayed.
eject


DISPLAY-SCREEN SECTION.
2000-DISPLAY-SCREEN.
    MOVE 'SLOTSEL' TO FIELDNAME.
    CALL 'MPCUR' USING SCRN-NUMBER SCRN-RETCODE FIELDNAME.
    CALL 'SCRN1141' USING SCRN-NUMBER, SCRN-FIELDS, SCRN-KEY.
    MOVE 'EVAL-PFKEYS' TO STEP-INDICATOR.
EXIT-2000-DISPALY-SCREEN.
eject


EVALUATE-PFKEYS SECTION.
3000-EVALUATE-PFKEYS.
    EVALUATE TRUE
       WHEN RETURN-KEY
            PERFORM 3900-BAD-KEY
       WHEN PF1
            PERFORM 3100-CALL-HELP
       WHEN PF2
            PERFORM 3900-BAD-KEY
       WHEN PF3
            MOVE 'CPPO0014' TO NEXT-PROGRAM-ID
            MOVE 'CPPO0015' TO PREV-PROGRAM-ID
            MOVE 'XFER-CONTROL' TO STEP-INDICATOR
*   All moves will occur in 4000-Transfer-Control
       WHEN PF4
            IF PRC-OK = 'Y' THEN
               PERFORM 3400-PROCESS-SEL
               MOVE 'N' TO PRC-OK
            ELSE PERFORM 3900-BAD-KEY
            END-IF
       WHEN PF5
            PERFORM 3900-BAD-KEY
       WHEN PF6
            PERFORM 3900-BAD-KEY
       WHEN PF7
            IF PGB-OK = 'Y' THEN
               PERFORM 3700-PAGE-BCK
            ELSE PERFORM 3900-BAD-KEY
            END-IF
       WHEN PF8
            IF PGF-OK = 'Y' THEN
               PERFORM 3800-PAGE-FWD
            ELSE PERFORM 3900-BAD-KEY
            END-IF
       WHEN PF9
```

```
            IF CANCEL-OK = 'Y' THEN
               PERFORM 3999-CANCEL
               MOVE 'N' TO CANCEL-OK
            ELSE PERFORM 3900-BAD-KEY
            END-IF
       WHEN PF10
            PERFORM 3900-BAD-KEY
       WHEN PF11
            PERFORM 3900-BAD-KEY
       WHEN PF12
            PERFORM 3900-BAD-KEY
       WHEN OTHER
            PERFORM 3900-BAD-KEY
    END-EVALUATE.
EXIT-3000-EVALUATE-PFKEYS.
eject
3100-CALL-HELP.
     MOVE THIS-PROGRAM-ID TO PREV-PROGRAM-ID.
     MOVE 'CPPOHELP' TO NEXT-PROGRAM-ID.
*    CALL 'CPPOHELP' USING DFHCOMMAREA.
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-3100-CALL-HELP.

* Selection of an Interview Slot
3400-PROCESS-SEL.
     MOVE SLOTSEL TO SLOT-N.
     MOVE SLOT-X TO SR.


* Check for invalid selection
     IF SR  >  0 AND SR  < 14  THEN
        MOVE 'Y' TO CONT-LOGIC
        ADD NDX-START TO SR
            GIVING SUB1
        SUBTRACT 1 FROM SUB1
     ELSE MOVE SPACES TO SYSMSG
          MOVE 'Enter Slot # from 01,02,03...brightened' TO
               SYSMSG
          MOVE 'N' TO CONT-LOGIC
     END-IF.


*  Check for Slot already TAKEN
          IF SLOT-STATUS (SUB1) = 'B' AND CONT-LOGIC = 'Y' THEN
             MOVE STUDENT-SSN TO STU-SQL-STU-SSN
             PERFORM 90020-UPDATE-INTERVIE
             MOVE 'N' TO BID-STATUS
             PERFORM 90030-UPDATE-BID-STATUS
             PERFORM 90041-UPDATE-SLOTS-TAKEN
          ELSE
             MOVE SPACES TO SYSMSG
             MOVE 'The slot is already taken; Choose another.' TO
                  SYSMSG
             MOVE 'N' TO CONT-LOGIC
          END-IF.


          IF DB-STATUS-INDICATOR = 'DUP' GO TO 99000-BACKOUT
          ELSE
          IF DB-STATUS-INDICATOR = 'OK' AND CONT-LOGIC = 'Y'
             MOVE 'Y' TO EXIST-AT-ALL
             MOVE SLOT-ST-P TO STUD-SSN (SUB1)
             MOVE 'D' TO SLOT-STATUS (SUB1)
             MOVE TABLE-NUM (SUB1) TO EXIST-PAGE
             MOVE SUB1 TO EXIST-NDX
```

```
           MOVE SR  TO EXIST-SLOT
           MOVE I-DATE (SUB1) TO EXIST-DATE
           MOVE I-TIME (SUB1) TO EXIST-TIME
         PERFORM 7100-SET-SCR-ATTS-BY-LINE
         PERFORM 7500-FINISH-FILL
         END-IF.


    MOVE 'DISP-SCREEN' TO STEP-INDICATOR.

EXIT-3400-PROCESS-SEL.
eject
3700-PAGE-BCK.
     SUBTRACT 1 FROM PAGE-NUM.
     MOVE TOP-OF-PG-NDX (PAGE-NUM) TO NDX-START.
     ADD 12 NDX-START
         GIVING NDX-END.
     PERFORM 7100-SET-SCR-ATTS-BY-LINE.
     PERFORM 7500-FINISH-FILL.
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-3700-PAGE-BCK.

3800-PAGE-FWD.
     ADD 1 TO PAGE-NUM.
     MOVE TOP-OF-PG-NDX (PAGE-NUM) TO NDX-START.
     IF PAGE-NUM = MAX-PAGE THEN
        MOVE MAX-SUB1 TO NDX-END
     ELSE ADD 12 NDX-START
          GIVING NDX-END.
     PERFORM 7100-SET-SCR-ATTS-BY-LINE.
     PERFORM 7500-FINISH-FILL.
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
EXIT-3800-PAGE-FWD.
eject
3900-BAD-KEY.
     MOVE SPACES TO SYSMSG.
     MOVE 'Incorrect Key Pressed' TO SYSMSG.
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
 EXIT-3900-BAD-KEY.

3999-CANCEL.
     MOVE EXIST-NDX TO SUB1.
     MOVE 'X' TO BID-STATUS.
     PERFORM 90030-UPDATE-BID-STATUS.
     PERFORM 90042-UPDATE-SLOTS-TAKEN.
     MOVE SPACES TO STU-SQL-STU-SSN.
     PERFORM 90020-UPDATE-INTERVIE.
     MOVE 'N' TO EXIST-AT-ALL.
     MOVE ZERO TO EXIST-PAGE.
     MOVE ZERO TO EXIST-NDX.
     MOVE ZERO TO EXIST-SLOT.
     MOVE SPACES TO EXIST-DATE.
     MOVE SPACES TO EXIST-TIME.
     MOVE SLOT-ST-N TO STUD-SSN (SUB1).
     MOVE 'B' TO SLOT-STATUS (SUB1).
     PERFORM 7100-SET-SCR-ATTS-BY-LINE.
     PERFORM 7500-FINISH-FILL.
     MOVE 'DISP-SCREEN' TO STEP-INDICATOR.
 EXIT-3999-CANCEL.
 eject


 4000-TRANSFER-CONTROL SECTION.
*    Populate those COMMAREA fields needed by next program.
     MOVE THIS-PROGRAM-ID  TO PREV-PROGRAM-ID.
```

```
     PERFORM 90200-PUT-DFHCOMMAREA.
*    MOVE 0 TO RETURN-CODE.
 EXIT-TRANSFER-CONTROL.
 eject


 5000-FILL-SCREEN-ONE SECTION.
*  Student already signed up
 5100-FILL-SCREEN-EXIST.
      MOVE EXIST-PAGE TO PAGE-NUM.
      MOVE TOP-OF-PG-NDX (PAGE-NUM) TO NDX-START.
      IF PAGE-NUM = MAX-PAGE THEN
         MOVE MAX-SUB1 TO NDX-END
      ELSE ADD 12 TO NDX-START GIVING NDX-END.
 EXIT-5100-FILL-SCREEN-EXIST.

*  Student not/initial signed up
 5500-FILL-SCREEN-NOTEXIST.
      MOVE 1 TO PAGE-NUM.
      MOVE TOP-OF-PG-NDX (PAGE-NUM) TO NDX-START.
      IF MAX-PAGE = 1 THEN
         MOVE MAX-SUB1 TO NDX-END
      ELSE ADD 12 TO NDX-START GIVING NDX-END.
 EXIT-5500-FILL-SCREEN-NOTEXIST.
 eject



 7000-PROCESS-SCREEN-ONE SECTION.
*  Set each screen attribute/field
 7100-SET-SCR-ATTS-BY-LINE.
      ACCEPT MILITARY-DATE FROM DATE.
      MOVE YR TO STD-Y.
      MOVE MO TO STD-M.
      MOVE DY TO STD-D.
      MOVE D-STD TO SCRDATE.
      ACCEPT CURR-TIME FROM TIME.
      MOVE HH TO HR.
      MOVE MM TO MN.
      MOVE SS TO SC.
      MOVE SCR-TIME TO SCRTIME.
      MOVE  STUDENT-SSN TO STUDID.
      MOVE  SCHEDUL-EMP-NO TO EMPNUM.
      MOVE  SCHEDUL-SCH-NO TO SCHDNUM.
      MOVE  SQL-EMP-NAME TO EMPNAME.
      MOVE ZEROS TO SUB1.


* If more than 13 (1 page) of interviews then MORE is displayed
      IF MAX-SUB1 > 13
         MOVE 'MORSLOT' TO FIELDNAME
         MOVE 'PROT' TO PROT
         MOVE 'BRIGHT' TO BRIGHT
         CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE, FIELDNAME
                            PROT, BRIGHT
         MOVE 'MORE' TO MORSLOT
      END-IF.


      MOVE ZEROS TO SR.
      MOVE 1 TO SR.
      MOVE TABLE-NUM (NDX-START) TO PAGE-NUM.
```

```
     PERFORM 7150-VARYING VARYING SUB1
        FROM NDX-START BY 1 UNTIL SUB1 > NDX-END.


EXIT-7100-SET-SCR-ATTS-BY-LINE.
eject

7150-VARYING.
        PERFORM 7200-SET-FLD.
        IF SLOT-STATUS (SUB1) = 'D' THEN
           MOVE 'PROT' TO PROT
           MOVE 'DIM' TO BRIGHT
           CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                       FIELDNAME, PROT, BRIGHT
        ELSE
                MOVE 'PROT' TO PROT
                MOVE 'BRIGHT' TO BRIGHT
                CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                            FIELDNAME, PROT, BRIGHT
        END-IF.
        STRING
             I-SLOT-NUM (SUB1)
             ' '
             I-DATE (SUB1)
             ' '
             I-TIME (SUB1)
             ' '
             STUD-SSN (SUB1) DELIMITED BY SIZE
             INTO FLD-NAM (SR)
        END-STRING
        ADD 1 TO SR.


7200-SET-FLD.
      EVALUATE SR
         WHEN 1
              MOVE 'ROW01' TO FIELDNAME
         WHEN 2
              MOVE 'ROW02' TO FIELDNAME
         WHEN 3
              MOVE 'ROW03' TO FIELDNAME
         WHEN 4
              MOVE 'ROW04' TO FIELDNAME
         WHEN 5
              MOVE 'ROW05' TO FIELDNAME
         WHEN 6
              MOVE 'ROW06' TO FIELDNAME
         WHEN 7
              MOVE 'ROW07' TO FIELDNAME
         WHEN 8
              MOVE 'ROW08' TO FIELDNAME
         WHEN 9
              MOVE 'ROW09' TO FIELDNAME
         WHEN 10
              MOVE 'ROW10' TO FIELDNAME
         WHEN 11
              MOVE 'ROW11' TO FIELDNAME
         WHEN 12
              MOVE 'ROW12' TO FIELDNAME
         WHEN 13
              MOVE 'ROW13' TO FIELDNAME
         WHEN OTHER
              DISPLAY 'INCORRECT INDEXING OF SR VARIABLE'
              PERFORM 99000-BACKOUT
```

```
      END-EVALUATE.
 EXIT-7200-SET-FLD.


* All data fields have been initialized;now fill logic fields
 7500-FINISH-FILL.
      IF SR < 14 THEN
         MOVE 'PROT' TO PROT
         MOVE 'BRIGHT' TO BRIGHT
         PERFORM UNTIL SR > 13
           CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                FLD-NAM (SR), PROT, BRIGHT
                MOVE SPACES TO FLD-NAM (SR)
                ADD 1 TO SR
         END-PERFORM
      END-IF.


      MOVE FLD-NAM (01) TO ROW01.
      MOVE FLD-NAM (02) TO ROW02.
      MOVE FLD-NAM (03) TO ROW03.
      MOVE FLD-NAM (04) TO ROW04.
      MOVE FLD-NAM (05) TO ROW05.
      MOVE FLD-NAM (06) TO ROW06.
      MOVE FLD-NAM (07) TO ROW07.
      MOVE FLD-NAM (08) TO ROW08.
      MOVE FLD-NAM (09) TO ROW09.
      MOVE FLD-NAM (10) TO ROW10.
      MOVE FLD-NAM (11) TO ROW11.
      MOVE FLD-NAM (12) TO ROW12.
      MOVE FLD-NAM (13) TO ROW13.

      MOVE SPACES TO ACTMSG.
      eject
      IF EXIST-AT-ALL = 'Y'
           STRING
                'You are currently scheduled for: '
                EXIST-DATE
                ' '
                EXIST-TIME  DELIMITED BY SIZE
                INTO SYSMSG
           END-STRING
      END-IF.


      IF EXIST-AT-ALL = 'Y' AND PAGE-NUM = EXIST-PAGE
           MOVE EXIST-SLOT TO SLOT-X
           MOVE SLOT-N TO SLOTSEL
         IF OK-DATE = 'Y' THEN
           MOVE 'PF9;CANCEL INTERVIEW' TO PFFOUR
           MOVE 'Y' TO CANCEL-OK
         ELSE MOVE SPACES TO PFFOUR
              MOVE 'N' TO CANCEL-OK
      END-IF.


      IF EXIST-AT-ALL = 'Y' AND PAGE-NUM NOT EQUAL EXIST-PAGE
           MOVE SPACES TO SLOTSEL
           MOVE SPACES TO PFFOUR
           MOVE 'N' TO CANCEL-OK
      END-IF.
```

```
      IF EXIST-AT-ALL = 'N'
           STRING
                'Type slot number (2 digits) corresponding '
                'to Date & Time desired.' DELIMITED BY SIZE
                INTO ACTMSG
           END-STRING
          MOVE   'You are not scheduled for any interview.'
              TO SYSMSG
          MOVE 'PF4:PROCESS SELECTED TIME' TO PFFOUR
          MOVE '   ' TO SLOTSEL
          MOVE 'Y' TO PRC-OK
      END-IF.


      IF MAX-PAGE > 1  AND PAGE-NUM = 1 THEN
            MOVE SPACES TO PFSEVEN
            MOVE 'PFEIGHT' TO FIELDNAME
            MOVE 'PROT' TO PROT
            MOVE 'dim' TO BRIGHT
            CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                         FIELDNAME, PROT, BRIGHT
            MOVE 'PF8:PG FWD' TO PFEIGHT
            MOVE 'Y' TO PGF-OK
            MOVE 'N' TO PGB-OK
      END-IF.


      IF MAX-PAGE > 1 AND PAGE-NUM = MAX-PAGE THEN
            MOVE SPACES TO PFEIGHT
            MOVE 'PFSEVEN' TO FIELDNAME
            MOVE 'PROT' TO PROT
            MOVE 'dim' TO BRIGHT
            CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                         FIELDNAME, PROT, bright
            MOVE 'PF7:PG BCK' TO PFSEVEN
            MOVE 'N' TO PGF-OK
            MOVE 'Y' TO PGB-OK
      END-IF.


      IF MAX-PAGE > 1 AND PAGE-NUM NOT EQUAL 1 AND
         PAGE-NUM NOT EQUAL MAX-PAGE THEN
         MOVE 'PFSEVEN' TO FIELDNAME
         MOVE 'PROT' TO PROT
         MOVE 'DIM' TO BRIGHT
         CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                      FIELDNAME, PROT, BRIGHT
         MOVE 'PF7:PG BCK' TO PFSEVEN
         MOVE 'PFEIGHT' TO FIELDNAME
         MOVE 'PROT' TO PROT
         MOVE 'DIM' TO BRIGHT
         CALL 'MFSET' USING SCRN-NUMBER, SCRN-RETCODE,
                      FIELDNAME, PROT, BRIGHT
         MOVE 'PF8:PG FWD' TO PFEIGHT
         MOVE 'Y' TO PGF-OK
         MOVE 'Y' TO PGB-OK
      END-IF.


      IF MAX-PAGE = 1
         MOVE SPACES TO PFSEVEN
```

```
         MOVE SPACES TO PFEIGHT
         MOVE 'N' TO PGF-OK
         MOVE 'N' TO PGB-OK
      END-IF.

      MOVE 'PF1:HELP' TO PFONE.
      MOVE 'PF3:QUIT' TO PFTHREE.

 EXIT-7500-FINISH-FILL.
 eject
 70000-BRANCH-LOGIC SECTION.
*  Initial filling of 'onboard' table from SQL tables
 77000-TABLE-FILL.
      MOVE 'N' TO EXIST-AT-ALL.
      EXEC SQL DECLARE C1 CURSOR FOR
      SELECT INT_DATE,INT_TIME,SSN,INT_NO
*INT_ROOM TAKEN OUT 10/14/92 S.PETER
      FROM INTERVIEW
           WHERE EMP_NO = :SQL-INTERVIE-EMP-NO
           AND   SCH_NO = :SQL-INTERVIE-SCH-NO
           ORDER BY INT_NO,INT_DATE,INT_TIME
*          ORDER BY INT_NO,INT_DATE,INT_TIME
           END-EXEC.
       IF SQLCODE = 0 THEN
          MOVE 'OK' TO DB-STATUS-INDICATOR
       ELSE IF SQLCODE = 100
          MOVE 'EOF' TO DB-STATUS-INDICATOR
       ELSE GO TO 99200-DB-ABEND
       END-IF.
       EXEC SQL OPEN C1 END-EXEC.
       MOVE ZEROS TO SUB1.
       PERFORM 77100-LOOP-FETCH VARYING SUB1 FROM 1 BY 1 UNTIL
          SUB1 > 100  OR SQLCODE = 100.
       MOVE ZEROS TO SUB1.

*  Julian-Date logic (for Last-Day Cancellation check)
       IF EXIST-NDX NOT EQUAL ZERO THEN
          MOVE I-DATE (EXIST-NDX) TO D-STD
       ELSE MOVE I-DATE (001) TO D-STD.
       MOVE STD-D TO DDD-X.
       PERFORM 77700-FIND-JULIAN.
       MOVE JUL-DATE TO L-JUL.
       ACCEPT MILITARY-DATE FROM DATE.
       MOVE MO TO STD-M.
       MOVE DY TO DDD-X.
       PERFORM 77700-FIND-JULIAN.
       MOVE JUL-DATE TO C-JUL.
         IF C-JUL > L-JUL THEN
            ADD 365 TO L-JUL.
       SUBTRACT C-JUL FROM L-JUL
                GIVING DAY-DIFF.
         IF DAY-DIFF = 1 THEN
            ACCEPT CURR-TIME FROM TIME
              IF HH > 09 THEN
                 MOVE 'N' TO OK-DATE
              ELSE MOVE 'Y' TO OK-DATE
              END-IF
         ELSE MOVE 'Y' TO OK-DATE.



  eject
```

```
77100-LOOP-FETCH.
     EXEC SQL FETCH C1
     INTO :SQL-INTERVIE-DATE,
          :SQL-INTERVIE-TIME,
*         :SQL-INTERVIE-ROOM:NUL-ROOM-NO,
          :SQL-INTERVIE-SSN:NUL-IND-SSN,
*added following field. S.Peter 12/20/92.
          :SQL-INTERVIE-INTNO
          END-EXEC.

     IF SQLCODE NOT EQUAL 100 THEN PERFORM 77600-TAB-LOGIC.
 eject

 77500-TABLE-CONTROL SECTION.

 77600-TAB-LOGIC.

       MOVE SUB1 TO TABLE-NDX (SUB1).

       DIVIDE SUB1 BY 13
              GIVING TABLE-NUM (SUB1)
                     REMAINDER SLOT-X.


       IF SLOT-X = 0 THEN MOVE '13' TO I-SLOT-NUM (SUB1)
          ELSE
       MOVE SLOT-N TO I-SLOT-NUM (SUB1)
       ADD 1 TO TABLE-NUM (SUB1).
       MOVE SQL-INTERVIE-DATE TO WS-DATE.
       MOVE D-YEAR TO STD-Y.
       MOVE D-MTH TO STD-M.
       MOVE D-DAY TO STD-D.
       MOVE D-STD TO I-DATE (SUB1).
       MOVE SQL-INTERVIE-TIME TO WS-TIME.
       MOVE T-HOUR TO STD-HR.
       MOVE T-MIN TO STD-MN.
       MOVE T-STD TO I-TIME (SUB1).
*added following line 12/20/92. S.Peter.
       MOVE SQL-INTERVIE-INTNO TO I-INTNO (SUB1).


*      IF NUL-ROOM-NO     EQUAL 0 THEN
*      MOVE SQL-INTERVIE-ROOM TO I-ROOM (SUB1)
*      ELSE MOVE SPACES TO I-ROOM (SUB1)
*      END-IF


       IF  NUL-IND-SSN NOT EQUAL 0 OR
        SQL-INTERVIE-SSN     EQUAL '          ' THEN
          MOVE 'B' TO SLOT-STATUS (SUB1)
          MOVE SLOT-ST-N TO STUD-SSN (SUB1)
       END-IF.


       IF SQL-INTERVIE-SSN = STUDENT-SSN AND
          EXIST-AT-ALL = 'N'  THEN
       MOVE 'Y' TO EXIST-AT-ALL
       MOVE SUB1 TO EXIST-NDX
       MOVE TABLE-NUM (SUB1) TO EXIST-PAGE
       MOVE SLOT-X TO EXIST-SLOT
       MOVE I-DATE (SUB1) TO EXIST-DATE
       MOVE I-TIME (SUB1) TO EXIST-TIME


       MOVE 'D' TO SLOT-STATUS (SUB1)
       MOVE SLOT-ST-P TO STUD-SSN (SUB1)
       MOVE SPACES TO SQL-INTERVIE-SSN
       END-IF.


     IF NUL-IND-SSN     EQUAL 0 AND
        SQL-INTERVIE-SSN NOT EQUAL '          ' THEN
          MOVE 'D' TO SLOT-STATUS (SUB1)
     MOVE SLOT-ST-Y TO STUD-SSN (SUB1).
     MOVE 0 TO NUL-IND-SSN.
     MOVE SUB1 TO MAX-SUB1.
     MOVE TABLE-NUM (SUB1) TO MAX-PAGE.


 EXIT-77600-TAB-LOGIC.

 eject

 77700-FIND-JULIAN.

     MOVE ZEROS TO JUL-DATE.
     EVALUATE STD-M
        WHEN 01
                ADD DDD TO JUL-DATE
        WHEN 02
                ADD 31 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 03
                ADD 59 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 04
                ADD 90 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 05
                ADD 120 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 06
                ADD 151 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 07
                ADD 181 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 08
                ADD 212 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 09
                ADD 243 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 10
                ADD 273 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 11
                ADD 304 TO JUL-DATE
                ADD DDD TO JUL-DATE
        WHEN 12
                ADD 334 TO JUL-DATE
                ADD DDD TO JUL-DATE
     END-EVALUATE.
 EXIT-77700-FIND-JULIAN.
 eject


 90000-SQL-ROUTINE SECTION.
```

```
 90010-GET-SELECTED-EMP.                                                EMP00010
     EXEC SQL                                                           EMP00020
          SELECT NAME INTO                                              EMP00030
            :SQL-EMP-NAME:NUL-IND-EMPNAME                               EMP00080
          FROM EMPLOYER                                                 EMP00120
          WHERE EMP_NO = :SQL-EMP-EMP-NO                                EMP00130
     END-EXEC.                                                          EMP00140
     IF SQLCODE = 0 THEN                                                EMP00150
        MOVE 'OK' TO DB-STATUS-INDICATOR
     ELSE IF SQLCODE = 100
        MOVE 'NOT FOUND' TO DB-STATUS-INDICATOR
     END-IF.
*MOVE INDIVIDUAL HOST VARIABLES INTO A RECORD STRUCTURE.                 EMP00160
      MOVE SQL-EMP-NAME  TO EMP-NAME.                                   EMP00210
 eject


 90020-UPDATE-INTERVIE.
*MOVE RECORD VARIABLES TO HOST VARIABLES.
     MOVE SCHEDUL-EMP-NO           TO  SQL-INTERVIE-EMP-NO.
     MOVE SCHEDUL-SCH-NO           TO  SQL-INTERVIE-SCH-NO.
     MOVE I-DATE (SUB1) TO D-STD.
     MOVE STD-M TO D-MTH.
     MOVE STD-D TO D-DAY.
     MOVE STD-Y TO D-YEAR.
     MOVE '-' TO VIRGULE-1.
     MOVE '-' TO VIRGULE-2.
     MOVE WS-DATE-X                TO  SQL-INTERVIE-DATE.
     MOVE I-TIME (SUB1) TO T-STD.
     MOVE STD-HR TO T-HOUR.
     MOVE STD-MN TO T-MIN.
     MOVE '.' TO VIRGULE-3.
     MOVE '.00  ' TO VIRGULE-4.
     MOVE WS-TIME-X                TO  SQL-INTERVIE-TIME.
*    MOVE I-ROOM (SUB1)            TO  SQL-INTERVIE-ROOM.
     MOVE STU-SQL-STU-SSN          TO  SQL-INTERVIE-SSN.
     MOVE I-INTNO (SUB1)           TO  SQL-INTERVIE-INTNO.

     EXEC SQL
          UPDATE INTERVIEW SET SSN = :SQL-INTERVIE-SSN,
                 STATUS = 'TAKEN   '
            WHERE EMP_NO = :SQL-INTERVIE-EMP-NO AND
                  SCH_NO = :SQL-INTERVIE-SCH-NO AND
                  INT_DATE = :SQL-INTERVIE-DATE AND
                  INT_TIME = :SQL-INTERVIE-TIME AND
                  INT_NO   = :SQL-INTERVIE-INTNO
*                 INT_ROOM = :SQL-INTERVIE-ROOM
     END-EXEC.
     IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR
     ELSE IF SQLCODE = -803
        MOVE 'DUP' TO DB-STATUS-INDICATOR
     ELSE GO TO 99200-DB-ABEND
     END-IF.
 eject




 90030-UPDATE-BID-STATUS.
     MOVE BID-STATUS TO SQL-BID-STATUS.
     MOVE STU-SQL-STU-SSN TO SQL-BID-SSN.
     MOVE SCHEDUL-EMP-NO TO SQL-BID-EMP-NO.
```

```
     MOVE SCHEDUL-SCH-NO TO SQL-BID-SCH-NO.

     EXEC SQL
          UPDATE BID SET STATUS = :SQL-BID-STATUS
             WHERE EMP_NO = :SQL-BID-EMP-NO  AND
                   SCH_NO = :SQL-BID-SCH-NO  AND
                   SSN    =:SQL-BID-SSN
     END-EXEC.

     IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR
     ELSE IF SQLCODE = -803
        MOVE 'DUP' TO DB-STATUS-INDICATOR
     ELSE GO TO 99200-DB-ABEND
     END-IF.
 EXIT-99030-UPDATE-BID-STATUS.
 eject

* NOT SO SURE WE NEED TO KEEP SLOTS_TAKEN DATA
* COLUMN DOES NOT EXIST IN SCHEDUL TABLE 09/27/92 S.PETER
* IF INCLUDE, UNCOMMENT PERFORM STMTS.
 90041-UPDATE-SLOTS-TAKEN.
     MOVE SCHEDUL-EMP-NO TO SQL-SCHEDUL-EMP-NO.
     MOVE SCHEDUL-SCH-NO TO SQL-SCHEDUL-SCH-NO.

     EXEC SQL
          UPDATE VISIT SET NUM_SLOTS_TAKEN = NUM_SLOTS_TAKEN + 1
                       WHERE EMP_NO = :SQL-SCHEDUL-EMP-NO AND
                             SCH_NO = :SQL-SCHEDUL-SCH-NO
     END-EXEC.

     IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR
     ELSE IF SQLCODE = -803
        MOVE 'DUP' TO DB-STATUS-INDICATOR
     ELSE GO TO 99200-DB-ABEND
     END-IF.
 EXIT-90041-UPDATE-SLOTS-TAKEN.
 eject

 90042-UPDATE-SLOTS-TAKEN.
     MOVE SCHEDUL-EMP-NO TO SQL-SCHEDUL-EMP-NO.
     MOVE SCHEDUL-SCH-NO TO SQL-SCHEDUL-SCH-NO.

     EXEC SQL
          UPDATE VISIT SET NUM_SLOTS_TAKEN =  NUM_SLOTS_TAKEN - 1
                 WHERE  EMP_NO = :SQL-SCHEDUL-EMP-NO  AND
                        SCH_NO = :SQL-SCHEDUL-SCH-NO
     END-EXEC.

     IF SQLCODE = 0 THEN
        MOVE 'OK' TO DB-STATUS-INDICATOR
     ELSE IF SQLCODE = -803
        MOVE 'DUP' TO DB-STATUS-INDICATOR
     ELSE GO TO 99200-DB-ABEND
     END-IF.
 EXIT-90042-UPDATE-SLOTS-TAKEN.
 eject

 90100-GET-DFHCOMMAREA SECTION.
*    MOVE '999' TO COMM-EMP-NUM.
*    MOVE '1' TO COMM-SCH-NUM.
*    MOVE '222222222' TO COMM-STUDENT-SSN.

     MOVE COMM-EMP-NUM TO SCHEDUL-EMP-NO.
```

```
     MOVE SCHEDUL-EMP-NO TO SQL-INTERVIE-EMP-NO.
     MOVE SCHEDUL-EMP-NO TO SQL-EMP-EMP-NO.
     MOVE COMM-SCH-NUM TO SCHEDUL-SCH-NO.
     MOVE SCHEDUL-SCH-NO TO SQL-INTERVIE-SCH-NO.
     MOVE COMM-STUDENT-SSN TO STUDENT-SSN.
     MOVE STUDENT-SSN TO STU-SQL-STU-SSN.
     MOVE 'PREP-SCREEN' TO STEP-INDICATOR.
 EXIT-GET-DFHCOMMAREA-EXIT.

 90200-PUT-DFHCOMMAREA SECTION.
     MOVE STUDENT-SSN TO COMM-STUDENT-SSN.
*    MOVE EXIST-DATE TO COMM-INTERVIEW-DATE.
*    MOVE EXIST-TIME TO COMM-INTERVIEW-TIME.
 EXIT-PUT-DFHCOMMAREA.
 eject


 99000-ABNORMAL-TERMINATION SECTION.
*   All abnormal terminations handled from here.


 99200-DB-ABEND.                                                          0
*****************************************************************  00623000
* THE FOLLOWING ROUTINE PRINTS THE SQLCA STRUCTURE:             *  00624000
*                                                               *  00625000
*  - SQLCODE = SQL RETURN CODE                                  *  00626000
*  - SQLERRM = SQL ERROR MESSAGE                                *  00627000
*  - SQLERRP = MODULE DETECTING ERROR                           *  00628000
*  - SQLERRD = INTERNAL ERROR VALUES                            *  00629000
*  - SQLWARN = SQL WARNING STRUCTURE                            *  00630000
*                                                               *  00631000
*****************************************************************  00632000
       DISPLAY '*************************************' UPON CONSOLE.  00633000
       DISPLAY '* PROGRAM ERROR ROUTINE ENTERED *' UPON CONSOLE.     00634000
       DISPLAY '* CHECK SYSPRINT FOR ERROR CODES*' UPON CONSOLE.     00635000
       DISPLAY '* CHANGES WILL BE BACKED OUT    *' UPON CONSOLE.     00636000
       DISPLAY '*************************************' UPON CONSOLE.  00637000
       MOVE SQLCODE TO DECODED-SQLCODE.                              00638000
       DISPLAY 'PROGRAM ERROR ROUTINE ENTERED'.                      00639000
       DISPLAY '              *******'.                              00640000
       DISPLAY 'A PROBLEM HAS BEEN DETECTED IN THE '.                00641000
       DISPLAY STEP-INDICATOR, ' PARAGRAPH.'.                        00642000
       DISPLAY 'THE FOLLOWING ERROR CODES SHOULD AID YOU IN'.        00643000
       DISPLAY 'PROBLEM DETERMINATION OF THE SQL STATEMENT.'.        00644000
       DISPLAY '              *******'.                              00645000
       DISPLAY 'SQLCODE : ' DECODED-SQLCODE.                         00646000
       DISPLAY 'SQLERRM : ' SQLERRMC.                                00647000
       DISPLAY 'SQLERRP : ' SQLERRP.                                 00648000
       PERFORM ERRD VARYING INDX2 FROM 1 BY 1 UNTIL INDX2 = 7.       00649000
       IF SQLWARN0 NOT EQUAL 'W'                                     00650000
           THEN GO TO 99000-BACKOUT,                                 00651000
           ELSE DISPLAY 'SQLWARN0: ' SQLWARN0,                       00652000
                DISPLAY 'SQLWARN1: ' SQLWARN1,                       00653000
                DISPLAY 'SQLWARN2: ' SQLWARN2,                       00654000
                DISPLAY 'SQLWARN3: ' SQLWARN3,                       00655000
                DISPLAY 'SQLWARN4: ' SQLWARN4,                       00656000
                DISPLAY 'SQLWARN5: ' SQLWARN5,                       00657000
                DISPLAY 'SQLWARN6: ' SQLWARN6,                       00658000
                DISPLAY 'SQLWARN7: ' SQLWARN7,                       00659000
                DISPLAY 'SQLWARN8: ' SQLWARN8,                       00660000
                DISPLAY 'SQLWARN9: ' SQLWARN9,                       00661000
                DISPLAY 'SQLWARNA: ' SQLWARNA,                       00662000
                GO TO 99000-BACKOUT.                                 00663000

 ERRD.
       MOVE SQLERRD (INDX2) TO DECODED-SQLERRD (INDX2).               0
```

```
       DISPLAY 'SQLERRD', INDX2, ': ', DECODED-SQLERRD (INDX2).   00666000
 EXIT-ERROR-ABEND.                                                  0621000
 eject


 99000-BACKOUT SECTION.                                           00667000
*****************************************************************  00668000
* 'WHENEVER' RESET TO 'CONTINUE' IN THE EVENT THAT THE ROLLBACK *  00669000
* WORK STATEMENT FAILS TO AVOID LOOP IN ERROR ROUTINE.          *  00670000
*****************************************************************  00671000
                                                                   00672000
      MOVE 'ABEND - BACKING OUT' TO STEP-INDICATOR.                 0674000
                                                                   00675000
      EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.                00676000
      EXEC SQL ROLLBACK WORK END-EXEC.                             00677000
      STOP RUN.
```