FINAL REPORT

UNIVERSITY OF MICHIGAN LAUNDRY SERVICES
SCHEDULING PROJECT

SPECIAL PROJECTS IN HOSPITAL SYSTEMS
IOE 481

by:
Neal A. Bloomfield
James M. Molloy
Gregory Scott

Dr. Richard J. Coffey
Winter Term 1989
Executive Summary

During January of 1989, a project for the University of Michigan Laundry Services was initiated by Dr. Richard Coffey, Director of Management Systems, based on the needs of Raymond Haggerty, Director of Alternative Revenue for the U of M hospital. The problem involved solving scheduling problems with laundry service employees. These problems consisted of worker complaints that they were not being given temporary promotions when they were entitled to them. These complaints led to the filing of grievances by the employees, which in turn caused a loss of valuable time for both workers and management. The grievance process was inconvenient and time consuming for both parties involved, and the scheduling problems had led to friction between management and union members.

There are approximately sixty employees that receive daily assignments at the laundry facility. There are several different pay grades that an employee can receive for performing different jobs. Workers at lower pay grades are entitled to temporary job promotions when higher paid employees are absent. According to the union contract, ability to perform a job and seniority are the two criteria used to assign temporary promotions.

A computer program was needed to perform the daily job assignments and temporary promotions quickly and objectively. Our job was to create this program and ensure that employees unfamiliar with computers would
be able to use it on a daily basis. The program also had to be able to accommodate employee and job modifications.

The program was written using the dBase III+ language and is currently in use at the laundry services. At the present time, management is very pleased with it's performance and ease of operation.

Neal Bloomfield
Jim Molloy
Greg Scott
April 1989
PURPOSE

The purpose of this project was to design and implement a software aided computer program that would assign laundry service employees to their appropriate work stations without any subjectivity on the part of management. These assignments were to be based upon union rules regarding daily job assignments and temporary promotions. The union contract states that priorities to be used as decision criteria are task ability, seniority, and personal preference.

BACKGROUND

The University of Michigan Laundry Services was having difficulty with employee grievances. These grievances stemmed from what employees felt was improper assignment of temporary promotions. Supervisors were being accused of awarding temporary promotions to employees they preferred rather than according to union rules. Both management and union workers were losing valuable time in a grievance process that arose from these disputes. The implementation of a computer program was suggested to eliminate the possibility of subjective decisions. Our project team was assigned the task of writing this program and putting it to use at the laundry facility.
**APPROACH**

The software that the program was designed on was a database management package called dBase 3+. It was necessary that the computer program have the capability to assign employees to their proper positions on a daily basis, add employees, delete employees, change employee job preferences and classifications, and print out all results.

The program enables the user to enter employees as present or absent and in return be provided with a printout that lists appropriate job assignments. These include temporary job promotions that are assigned in accordance with union rules. The assignment process is carefully documented in the copy of the computer program provided in the Appendix.

All of the functions provided by the computer program are entirely menu driven. If an employee takes a leave of absence or quits, they can easily be deleted from the database. New employees or changes within present employees and their classifications can also be easily manipulated through the menu system. All functions can be performed by a computer novice.
CURRENT SITUATION

The laundry service is currently using the computer program that was designed by our project team. Due to changes in the output format that were suggested by Chris Brooks, the output of daily job assignments is being used to help with payroll calculations. In recent weeks, supervisors have been trained to use the computer for early morning job assignments. The program has been constantly improved through their suggestions. Changes in the system at this point consist of format modifications. Format of printouts is being changed to accommodate the whims of managers and supervisors. Many suggestions were derived from a training session that involved team members, Chris Brooks, Raymond Haggerty and four laundry supervisors. All of these people were provided with a personal instruction manual so they could write notes about concepts they did not understand.

FINDINGS AND CONCLUSIONS

With the implementation of the scheduling program at the laundry facility, the disputes over assignments of temporary jobs should be eliminated. Job assignments are always completed in strict accordance with union rules, and thus no disputes may arise. The program evolved from a simple scheduling program into an aid for weekly payroll and a basis for a future payroll system. The final program resulted from constant
interaction between Chris Brooks and the project team through which many new ideas and features for the program were developed.

As a result of this project, many disputes between union members and management will be eliminated. Chris Brooks estimated that the yearly savings from eliminated grievances in monetary terms will be $1000. This figure was arrived at by calculating worker and management hours lost during the grievance process. Mr. Brooks mentioned, however, that the real value of the project would be to eliminate friction between management and union employees. He feels that no price tag can be placed on the worker satisfaction that will result from this project.

**RECOMMENDATIONS**

We would recommend that the program be expanded to more systematically deal with the tracking of payroll. One possibility would be to add to the program and give it the capability of calculating hours worked by an employee in each respective pay grade. The computer operator could simply enter what jobs they sent each employee to after running the scheduling program. The computer would automatically have the jobs entered as they resulted from the running of the scheduling program. The operator would only have to enter changes that occurred during the day. In this way, weekly pay may be quickly and easily tracked. It may be necessary
at a future point to add additional job classifications to accommodate the fluctuating structure of the laundry service facility. All programming was carefully commented in case changes such as these are needed in the future.

All aspects of this project that our group is connected with have been implemented and are currently in use. The above recommendations for expansion of the payroll aspects of our program have been suggested to Chris Brooks and are currently under consideration. Decisions regarding these suggestions must be made by Chris Brooks and Raymond Haggerty and do not fall within the scope of our project.
SCHPROG1.PRG
THIS SECTION PUTS THE EMPLOYEES IN THEIR OWN JOB CLASSIFICATION
UMH LAUNDRY SERVICES
****replacing all scores with 99 clears the 'memory' of jobs
REPLACE ALL SCORE WITH 99
GO TOP
**** this section puts looks to see what the employees job
**** classification is and then puts them into thier job
**** the first comparison will be used as an example
DO WHILE .T.
  IF EOF()
    EXIT
  ENDIF
**** if the classification is feeder folder primary
  IF UPPER(CLASSIFIC)= 'FFPRIMARY' .AND. NFFPRIMARY<>0 .AND. PRESENT
**** replace the number of ffprimary workers one less (nffprimary)
    STORE (NFFPRIMARY-1) TO NFFPRIMARY
**** let the score reflect that persons current job
**** hence score = the preference ranking for their job
    REPLACE SCORE WITH FFPRIMARY
**** go to the next employee
    SKIP
**** go back to the top and search for the job of the next employee
  LOOP
ENDIF
**** the following jobs are searched in the same fashion as above
  IF UPPER(CLASSIFIC)= 'FFNUMTHREE' .AND. NFFNUMTHREE<>0 .AND. PRESENT
    STORE (NFFNUMTHREE-1) TO NFFNUMTHREE
    REPLACE SCORE WITH FFNUMTHREE
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'FFSWING' .AND. NFFSWING<>0 .AND. PRESENT
    STORE (NFFSWING-1) TO NFFSWING
    REPLACE SCORE WITH FFSWING
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'FFMACHFOLD' .AND. NFFMACHFOLD<>0 .AND. PRESENT
    STORE (NFFMACHFOLD-1) TO NFFMACHFOLD
    REPLACE SCORE WITH FFMACHFOLD
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'FFMISC' .AND. NFFMISC>0 .AND. PRESENT
    STORE (NFFMISC-1) TO NFFMISC
    REPLACE SCORE WITH FFMISC
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'CLSOILSORT' .AND. NCLSOILSORT<>0 .AND. PRESENT
    STORE (NCLSOILSORT-1) TO NCLSOILSORT
    REPLACE SCORE WITH CLSOILSORT
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'CLCAMPUS' .AND. NCLCAMPUS<>0 .AND. PRESENT
    STORE NCLCAMPUS -1 TO NCLCAMPUS
    REPLACE SCORE WITH CLCAMPUS
    SKIP
    LOOP
ENDIF
  IF UPPER(CLASSIFIC)= 'CLBASKET' .AND. NCLBASKET>0 .AND. PRESENT
    STORE NCLBASKET -1 TO NCLBASKET
    REPLACE SCORE WITH CLBASKET
    SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='CLFLOOR' .AND. NCLFLOOR<>0 .AND. PRESENT
STORE NCLFLOOR -1 TO NCLFLOOR
REPLACE SCORE WITH CLFLOOR
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='CUSTODIAN' .AND. NCUSTODIAN<>0 .AND. PRESENT
STORE NCUSTODIAN -1 TO NCUSTODIAN
REPLACE SCORE WITH CUSTODIAN
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='STOCKKEEP' .AND. NSTOCKKEEP<>0 .AND. PRESENT
STORE NSTOCKKEEP -1 TO NSTOCKKEEP
REPLACE SCORE WITH STOCKKEEP
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='CHECKHOSP' .AND. NCHECKHOSP>0 .AND. PRESENT
STORE (NCHECKHOSP -1) TO NCHECKHOSP
REPLACE SCORE WITH CHECKHOSP
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='CHECKTRAYS' .AND. NCHECKTRAYS<>0 .AND. PRESENT
STORE NCHECKTRAYS -1 TO NCHECKTRAYS
REPLACE SCORE WITH CHECKTRAYS
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='CHECKCAMPU' .AND. NCHECKCAMPU<>0 .AND. PRESENT
STORE NCHECKCAMPU -1 TO NCHECKCAMPU
REPLACE SCORE WITH CHECKCAMPU
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='WL' .AND. NWL<>0 .AND. PRESENT
STORE NWL -1 TO NWL
REPLACE SCORE WITH WL
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='OLM' .AND. NOLM<>0 .AND. PRESENT
STORE NOLM -1 TO NOLM
REPLACE SCORE WITH OLM
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='OMV' .AND. NOMV>0 .AND. PRESENT
STORE NOMV -1 TO NOMV
REPLACE SCORE WITH OMV
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='MACHREP' .AND. NMACHREP<>0 .AND. PRESENT
STORE NMACHREP -1 TO NMACHREP
REPLACE SCORE WITH MACHREP
SKIP
LOOP
ENDIF
IF UPPER(CLASSIFIC)='GARMENT' .AND. NGARMENT<>0 .AND. PRESENT
STORE NGARMENT -1 TO NGARMENT
REPLACE SCORE WITH GARMENT
SKIP
LOOP
ENDDIF
IF UPPER(CLASSIFIC)='PRESSOP' .AND. NPRESSOP<>0 .AND. PRESENT
STORE NPRESSOP -1 TO NPRESSOP
REPLACE SCORE WITH PRESSOP
SKIP
LOOP
ENDIF
SKIP

**** at the end of the database exit this loop
ENDDO

**** return to the last subroutine
RETURN

*** SCHEDUALING PROGRAM, UNIVERSITY OF MICHIGAN LAUNDRY SERVICES
*** LAUNDRY.PRG
*** this program allows the user to enter the day of the week
*** setting talk and echo off are necessary for this program
*** to function visibly well
SET TALK OFF
SET ECHO OFF
SET SCOREBOARD OFF
SET STATUS OFF

**** this is the database
USE SCHEDULE.DBF
**** this resets the attendance so that currently everyone
**** is marked present
replace all present with .t.
**** initialize choice
STORE '' TO CHOICE
DO WHILE .T.

**** do initial menu to enter the day of the week or SPECIAL
CLEAR
@1,0 TO 19,79
@2,19 SAY **** LAUNDRY SERVICES SCHEDUALING PROGRAM ****
@3,1 TO 3,78
@17,1 TO 17,78
@7,10 SAY 'Please enter the day of the week.....'
@8,15 SAY 'MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY'
@11,10 SAY 'Note...' 
@12,15 SAY 'If there are many workers absent, and you would'
@13,15 SAY 'like to change the normal number of jobs, enter'
@14,15 SAY 'SPECIAL.'
@17,5 SAY ''

***** the accept command prompts the user and stores the
***** keyboard entry to the variable day
ACCEPT 'I' TO DAY

***** make the day of the week all capitals
STORE UPPER(DAY) TO DAY

***** initialize all variables so that they become public to all
***** subroutines throughout the program

***** these variables represent the # of employees working

***** at each job station
STORE 0 TO NFFPRIMARY
STORE 0 TO NFFNUMTHREE
STORE 0 TO NFFSWING
STORE 0 TO NFFMACHFOLD
STORE 0 TO NFFMISC
STORE 0 TO NCLSOILSORT
STORE 0 TO NCLCAMPUS
STORE 0 TO NCLBASKET
STORE 0 TO NCLFLOOR
STORE 0 TO NCUSTODIAN
STORE 0 TO NSTOCKKEEP
STORE 0 TO NCHECKHOSP
STORE 0 TO NCHECKTRAYS
STORE 0 TO NCHECKCAMPU
STORE 0 TO NWL
STORE 0 TO NOI M
STORE 0 TO NOV
STORE 0 TO NMACHREP
STORE 0 TO NGARMENT
STORE 0 TO NPRESSOP

*** check to see if the entry was valid if it is proceed
*** if it isn't then return to the beginning of this program
*** to reenter the day
IF DAY='MO' .OR. DAY='TU' .OR. DAY='WE' .OR. DAY='TH' .OR. DAY='FR' .OR. DAY='F'
DO SCHEDULE
SET SCOREBOARD ON
SET STATUS ON
RETURN
ELSE
LOOP
ENDIF
ENDDO
SET STATUS ON
SET SCOREBOARD ON
RETURN

*** addemply.prg
*** u of m hospital laundry services
*** this program adds people to the database
*** it takes into account not only totally new
*** employees, but also old employees that are
*** returning after a leave of absence
SET TALK OFF
SET SCOREBOARD OFF
USE SCHEDULE
CLEAR

*** initialize verf, namer, quite
/VERF=.T.
STORE SPACE(25) TO NAMER
STORE SPACE(12) TO QUITE

*** prompt the user to enter the seniority date to tempdate
@ 4, 8 TO 16, 65
@ 8, 13 SAY "PLEASE ENTER THE SENIORITY DATE OF THE EMPLOYEE"
@ 9, 13 SAY "YOU WOULD LIKE TO ADD TO THE LIST. ENTER MONTH"
@ 10, 13 SAY "FIRST, DAY SECOND, YEAR THIRD, AND PLACE SLASHES"
@ 11, 13 SAY "BETWEEN EACH. (e.g. 4/10/70 IS APRIL 4, 1970.)"
ACCEPT ’ ’ TO TEMPDAT E

*** look to see if any employee has the same date
LOCATE FOR SENIORITY=CTOD(TEMPDATE)
*** if some other employee has the same date
*** the end of file will not be reached
*** and the program enters this section
DO WHILE .NOT. EOF()
*** this section asks if the employee should be entered
*** before or after the person with the same seniority
CLEAR
@ 15,10 SAY "Another employee has the same starting date."
@ 16,10 SAY "Do you want to insert employee before this employee?(Y/N)"
@ 18,30 SAY ’ ’ GET VERF PICTURE "Y"
READ
*** if the employee is to be entered before
*** the program exits here to insert the person
IF VERF
SET FORMAT TO INSMEN
INSERT BEFORE
CLEAR
CLOSE FORMAT
RETURN
ENDIF
CLEAR
*** ask if the employee should be entered after the current
*** employee with the same seniority
  @ 16,10 SAY "Do you want to insert new employee after this employee?(Y/N)"
  @ 18,30 SAY NAMES
  @ 19,40 SAY ' ' GET VERF PICTURE "Y" READ
*** if the employee is to be entered after then do this if loop
  IF VERF
*** check to see if there are anymore people with
*** the same seniority
    CONTINUE
*** if it finds someone else with the same seniority
*** then ask before or after all over again
  IF .NOT. EOF()
    CLEAR
    LOOP
*** if nobody else is found with the same seniority
*** then enter them after the person in question
  ELSE
    SET FORMAT TO INSMEN
    INSERT
    CLEAR
    CLOSE FORMAT
    RETURN
  ENDF
  ENDF
*** if the user enters that the employee should
*** be entered neither before or after then an error
*** is generated and the program is exited
  CLEAR
  @10,10 SAY 'You must choose to enter the employee either'
  @11,15 say 'before or after'+names
  @19,10 say 'Exiting program. Hit any key to continue.'
  WAIT '' TO WAITS
  RETURN
ENDDO
*** if the program gets this far then the employee
*** is either new or does not have a seniority date that
*** matches somebody else's
*** insert them in the database
  DO TOP
  DO WHILE .NOT. EOF()
*** locate their position in the database
    IF CTOD(TEMPDATE)>SENIORITY
      SKIP
    ELSE
*** once you find their position in the database, insert them
      SET FORMAT TO INSMEN
      INSERT BEFORE
      CLEAR
      CLOSE FORMAT
      RETURN
    ENDF
  ENDDO
*** if the end of the database is reached
*** insert them at the end
  SET FORMAT TO INSMEN
  INSERT
  CLEAR
  CLOSE FORMAT
  RETURN
**** SENIORITY.PRG
**** PRINTS THE SENIORITY LISTING
**** U OF M HOSPITALS
CLEAR
**** print the heading
  @10,20 SAY 'Ready Printer. Hit any key to continue.'
wait ' ' to waits
GO TOP
SET PRINT ON
?' EMPLOYEE SENIORITY DATE LISTING'
?''
?"NAME: REG O.T. S/V DEDUCT E U F/H Temp Promot"?

**** this index creates an alphabetical listing
index on names to nameind.ndx
set index to nameind.ndx
**** print each person until the end is reached
DO WHILE .NOT. EOF()
?'+NAMES+'______ _____ _____ E U ____ ________' '+dI
?'
SKIP
ENDDO
?'
SET PRINT OFF
EJECT
close index
**** this deletes the index so that if an employee is added
**** or deleted that the index will change every time
delete file nameind.ndx
RETURN

***** JOBNUM1.PRG
***** JOBNUM1, JOBNUM2, AND SPECIAL ARE SUBROUTINES DESIGNED TO
***** TELL THE COMPUTER HOW MANY PEOPLE WORK EACH JOB.
PUBLIC ALL
***** each job number variable name is 'n'+classific'
***** hence the number of jobs for the feeder folders primary
***** iron (classification=ffprimary) is nffprimary=4
STORE 4 TO NFFPRIMARY
STORE 4 TO NFFNUMTHREE
STORE 4 TO NFFSWING
STORE 13 TO NFFMACHFOLD
STORE 5 TO NCLSOILSORT
STORE 1 TO NCLCAMPUS
STORE 1 TO NCLBASKET
STORE 1 TO NCLFLOOR
STORE 2 TO NCUSTODIAN
STORE 1 TO NSTOCKKEEP
STORE 4 TO NCHECKHOSP
STORE 1 TO NCHECKTRAYS
STORE 1 TO NCHECKCAMPU
STORE 3 TO NWL
STORE 1 TO NOLM
STORE 3 TO NOMV
STORE 2 TO NMACHREP
STORE 1 TO NGARMENT
STORE 1 TO NPRESSOP
RETURN

***** SPECIAL.PRG
***** JOBNUM1, JOBNUM2, AND SPECIAL ARE SUBROUTINES DESIGNED TO
***** TELL THE COMPUTER HOW MANY PEOPLE WORK EACH JOB.
***** this program prompts input for the number
***** of workers to work each job
CLEAR
1,1 SAY 'You entered that it was a SPECIAL day, and that you'
2,1 SAY 'wanted to change the number of employees at each station.'
3,1 SAY 'This section allows you to enter the number of assigned jobs.'
6,1 SAY 'JOBS | NORMAL | NEW'
7,1 SAY '--------------------------------------------'
PUBLIC ALL
**** each input statement results in a prompt for information
INPUT 'FF PRIMARY  |  4 | ' TO NFFPRIMARY
INPUT 'FF NUMTHREE|  4 | ' TO NFFNUMTHREE
INPUT 'FF SWING   |  4 | ' TO NFFSWING
INPUT 'FF MACHFOLD|  2 | ' TO NFFMACHFOLD
INPUT 'FF MISC    | 13 | ' TO NFFMISC
INPUT 'CL SOILSORT|  5 | ' TO NCCLSOILSORT
INPUT 'CL CAMPUS  |  1 | ' TO NCCLCAMPUS
INPUT 'CL BASKET  |  1 | ' TO NCCLBASKET
INPUT 'CL FLOOR   |  1 | ' TO NCCLFLOOR
INPUT 'CUSTODIAN  |  2 | ' TO NCUSTODIAN
INPUT 'STOCKKEEP  |  1 | ' TO NSTOCKKEEP
INPUT 'CHECKHOSP  |  4 | ' TO NCHECKHOSP
INPUT 'CHECKTRAYS |  1 | ' TO NCHECKTRAYS
INPUT 'CHECKCAMPUS|  1 | ' TO NCHECKCAMPUS
INPUT 'WL         |  3 | ' TO NWL
INPUT 'OLM        |  1 | ' TO NOLM
INPUT 'OMV        |  3 | ' TO NOMV
INPUT 'MACHREP    |  2 | ' TO NMACHREP
INPUT 'GARMENT    |  1 | ' TO NGARMENT
INPUT 'PRESSOP    |  1 | ' TO NPRESSOP

RETURN

***** JOBNUM2.PRG
***** JOBNUM1, JOBNUM2, AND SPECIAL ARE SUBROUTINES DESIGNED TO
***** TELL THE COMPUTER HOW MANY PEOPLE WORK EACH JOB.
***** this program stores the number of workers per job into
***** variables. Example: nffprimary=4 means that there
***** will be four spots on the feeder folder primary iron
STORE 4 TO NFFPRIMARY
STORE 4 TO NFFNUMTHREE
STORE 4 TO NFFSWING
STORE 2 TO NFFMACHFOLD
STORE 13 TO NFFMISC
STORE 5 TO NCCLSOILSORT
STORE 1 TO NCCLCAMPUS
STORE 1 TO NCCLBASKET
STORE 1 TO NCCLFLOOR
STORE 2 TO NCUSTODIAN
STORE 1 TO NSTOCKKEEP

***** THIS IS THE ONLY DIFFERENCE TO JOBNUM1
STORE 3 TO NCHECKHOSP

*****
STORE 1 TO NCHECKTRAYS
STORE 1 TO NCHECKCAMPUS
STORE 3 TO NWL
STORE 1 TO NOLM
STORE 3 TO NOMV
STORE 2 TO NMACHREP
STORE 1 TO NGARMENT
STORE 1 TO NPRESSOP
RETURN

***** ATTEN2.PRG
***** ATTENDANCE PROGRAM
***** UNIVERSITY OF MICHIGAN HOSPITAL LAUNDRY SERVICE
***** this program is accessed by attendan.prg program
***** all this does is prompt the reader to enter the
***** attendance of each employee
@6,48 GET PRESENT

***** present is a logical variable representing if the
***** employee is present or not
READ
SKIP
IF EOF()
   RETURN
ENDIF
@7,48 GET PRESENT
READ
SKIP
  IF EOF()
    RETURN
  ENDIF
@8,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@9,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@10,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@11,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@12,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@13,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@14,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@15,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@16,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@17,48 GET PRESENT
READ
SKIP
IF EOF()
  RETURN
ENDIF
@18,48 GET PRESENT
READ
*** all this program does is provide a menu for
*** choosing either to print the preferences listing
*** or to print the seniority/payroll listing
DO  WHILE  .T.
  menu start
    CLEAR
    01, 5 TO 19, 74  DOUBLE
    03, 6 TO 3, 73
    02, 25  SAY  'Printing employee information'
    09, 25  say  '<Q>...Quit'
    010, 25  say  '<S>...Seniority listing only'
    011, 25  say  '<D>...Database printout'
    018, 35  say  'Enter Choice'
    017, 6 TO 17, 73
  menu end
  wait  ''  to choice
  ***  enter choice of options to choice
  DO  CASE
    CASE  UPPER(CHOICE)='Q'
      RETURN
    CASE  UPPER(CHOICE)='S'
      *************seniority/payroll listing
      DO  SENIORITY
        LOOP
    CASE  UPPER(CHOICE)='D'
      DO  DATAPRINT
    *************print preferences of database
    LOOP
  OTHERWISE
    LOOP
ENDCASE
ENDDO@  1,  2  SAY  "Enter employee name (Last, First)"
@  2,  45  SAY  "Enter employee class"
@  2,  45  SAY  "it appears in the listing below."
@  3,  2  SAY  "NAME"  
@  3,  13  GET  SCHEDULE->NAMES
@  4,  45  SAY  "CLASSIFIC"  
@  4,  56  GET  SCHEDULE->CLASSIFIC
@  5,  2  SAY  "SENIORITY DATE"  
@  5,  18  GET  SCHEDULE->SENIORITY
@  7,  2  SAY  "Fill in preference data below. The favorite job should be ranked wit
@  8,  2  SAY  "1, etc.. Jobs that are not available to the employee should be rank
@  10,  2  SAY  "FFPRIMARY"
@  10,  14  GET  SCHEDULE->FFPRIMARY
@  10,  20  SAY  "FFNUMTHREE"
@  10,  32  GET  SCHEDULE->FFNUMTHREE
@  10,  37  SAY  "FFSWING"
@  10,  49  GET  SCHEDULE->FFSWING
@  10,  54  SAY  "FFMACHFOLD"
@  10,  66  GET  SCHEDULE->FFMACHFOLD
@  12,  2  SAY  "FFMISC"
@  12,  14  GET  SCHEDULE->FFMISC
@  12,  20  SAY  "CLSOILSORT"
@  12,  32  GET  SCHEDULE->CLSOILSORT
@  12,  37  SAY  "CLCAMPUS"
@  12,  49  GET  SCHEDULE->CLCAMPUS
@  12,  54  SAY  "CLBASKET"
GET SCHEDULE->CLBASKET.

SAY "CLFLOOR"

GET SCHEDULE->CLFLOOR

SAY "CUSTODIAN"

GET SCHEDULE->CUSTODIAN

SAY "STOCKKEEP"

GET SCHEDULE->STOCKKEEP

SAY "CHECKHOSP"

GET SCHEDULE->CHECKHOSP

SAY "CHECKTRAYS"

GET SCHEDULE->CHECKTRAYS

SAY "CHECKCAMPU"

GET SCHEDULE->CHECKCAMPU

SAY "WL"

GET SCHEDULE->WL

SAY "OLM"

GET SCHEDULE->OLM

SAY "OMV"

GET SCHEDULE->OMV

SAY "MACHREP"

GET SCHEDULE->MACHREP

SAY "GARMENT"

GET SCHEDULE->GARMENT

SAY "PRESSOP"

GET SCHEDULE->PRESSOP

TO 19, 78  DOUBLE

TO 6, 77

SAY 'Name: '+names

SAY "PLEASE FILL IN THE EMPLOYEES PREFERENCES IN THE SCREEN"

SAY "BELOW. ENTER A 1 FOR THE MOST DESIRABLE JOB, ETC. THE"

SAY "EMPLOYEES OWN JOB SHOULD BE RANKED AFTER ALL OTHERS"

SAY "THAT ARE IN A HIGHER PAY GRADE."

SAY "FFPRIMARY"

GET SCHEDULE->FFPRIMARY

SAY "FFNUMTHREE"

GET SCHEDULE->FFNUMTHREE

SAY "FFSWING"

GET SCHEDULE->FFSWING

SAY "FFMACHFOLD"

GET SCHEDULE->FFMACHFOLD

SAY "FFMISC"

GET SCHEDULE->FFMISC

SAY "CLSOILSORT"

GET SCHEDULE->CLSOILSORT

SAY "CLCAMPUS"

GET SCHEDULE->CLCAMPUS

SAY "CLBASKET"

GET SCHEDULE->CLBASKET

SAY "CLFLOOR"

GET SCHEDULE->CLFLOOR

SAY "CUSTODIAN"

GET SCHEDULE->CUSTODIAN

SAY "STOCKKEEP"

GET SCHEDULE->STOCKKEEP

SAY "CHECKHOSP"

GET SCHEDULE->CHECKHOSP

SAY "CHECKTRAYS"

GET SCHEDULE->CHECKTRAYS

SAY "CHECKCAMPU"

GET SCHEDULE->CHECKCAMPU

SAY "WL"

GET SCHEDULE->WL

SAY "OLM"

GET SCHEDULE->OLM

SAY "OMV"
@ 17, 22 SAY "MACHREP"
@ 17, 34 GET SCHEDULE->MACHREP
@ 17, 40 SAY "GARMENT"
@ 17, 52 GET SCHEDULE->GARMENT
@ 17, 58 SAY "PRESSOP"
! 17, 70 GET SCHEDULE->PRESSOP
@ 1, 0 TO 19, 74 DOUBLE
@ 7, 1 TO 7, 73

****DATAPRIN.PRG

***** PRINTS THE SENIORITY LISTING
***** U OF M HOSPITALS
**** this program prints the preferences listing of all people

CLEAR

***** make the user ready the printer
@10,20 SAY 'Ready Printer. Hit any key to continue.'

wait'' to wait

***** this index alphabetizes the listing
INDEX ON NAMES TO NAMEIND.NDX
SET INDEX TO NAMEIND.NDX
GO TOP

***** print the title
SET PRINT ON ?'EMPLOYEE PREFERENCES LISTING' ?'

DO WHILE .NOT. EOF()

***** this section prints all data stored for each person on the
***** database

?' '+NAMES+' Seniority date: '+DTOC(SENIORITY)+' Classification: '+CLASSIFIC
?' Preferences:

************the following are all preference variables
?' FFPRIORITY '+LTRIM(STR(FFPRIMARY))
?' FNUMTHREE '+LTRIM(STR(FFNUMTHREE))
?' FSWING '+LTRIM(STR(FFSWING))
?' FMAChFOLD '+LTRIM(STR(FFMACHFOLD))
?' FMISC '+LTRIM(STR(FFMISC))
?' CSLOILSORT '+LTRIM(STR(CLSOILSORT))
?' CLCAMPUS '+LTRIM(STR(CLCAMPUS))
?' CLBASKET '+LTRIM(STR(CLBASKET))
?' CLFLOOR '+LTRIM(STR(CLFLOOR))
?' CUSTODIAN '+LTRIM(STR(CUSTODIAN))
?' STOCKKEEP '+LTRIM(STR(STOCKKEEP))
?' CHECKHOSP '+LTRIM(STR(CHECKHOSP))
?' CHECKTRAYS '+LTRIM(STR(CHECKTRAYS))
?' CHECKCAMPUS '+LTRIM(STR(CHECKCAMPUS))
?' W L '+LTRIM(STR(WL))
?' OLM '+LTRIM(STR(OLM))
?' OMV '+LTRIM(STR(OMV))
?' MACHREP '+LTRIM(STR(MACHREP))
?' GARMENT '+LTRIM(STR(GARMENT))
?' PRESSOP '+LTRIM(STR(PRESSOP))

SKIP

***** continue if the end has not been reached
ENDDO

SET PRINT OFF
EJECT

***** get rid of the index so that it can be
***** re alphabetized if an employee is added or deleted
CLOSE INDEX
DELETE FILE NAMEIND.NDX
RETURN

*** delemplu
*** u of m hospital laundry services
*** this program deletes an employee from the database

CLEAR
SET TALK OFF
SET STATUS OFF
**** initialize namedel, verf
STORE SPACE(25) TO NAMEDEL
STORE 'N' TO VERF
CHECK=.T.
**** give a menu
USE SCHEDULE
GO TOP
@ 3, 15 SAY "**************************************************************************"  
@ 4, 15 SAY "*"  
@ 5, 15 SAY "*" EMPLOYEE DELETIONS ARE PERMANENT.  
@ 6, 15 SAY "*" IF AN EMPLOYEE IS ACCIDENTALLY DELETED,  
@ 7, 15 SAY "*" THEY MUST BE REENTERED USING THE ADD  
@ 8, 15 SAY "*" EMPLOYEE MENU OPTION.  
@ 9, 15 SAY "*"  
@ 10, 15 SAY "**************************************************************************"  
@ 1, 4 TO 18, 73

**** prompt the user to enter the name of the employee to be deletee
@ 14, 11 SAY "PLEASE ENTER THE LAST NAME OF THE EMPLOYEE TO BE DELETED"
ACCEPT ' ' TO TEMP

*** find that person in the database
LOCATE FOR UPPER(NAMES)=UPPER(TEMP) DO WHILE UPPER(VERF)='Y'.AND. .NOT. EOF()
  CLEAR
  @ 10,20 SAY 'Is the employee you want to delete? (Y/N)'
  @ 12,30 SAY NAMES
  @ 9,15 TO 13,64 DOUBLE
  @ 14,30 SAY " " GET VERF PICTURE "Y"
  READ
*** if this is not the person find people with similar
*** last names
  IF UPPER(VERF)='N'
    CONTINUE
  ENDF
ENDDO

*** if the person is not found give this error message
*** and exit the program
IF EOF()
  CLEAR
  @ 10,20 SAY "Employee not found. Please check spelling."
  @ 11,20 SAY "No other employee by that name was found."
  @ 13,20 SAY "Hit any key to continue."
  WAIT ' ' TO WAITS
  RETURN
ELSE

DELETE
**** prompt the user to make sure they want to delete the
**** employee in question
@ 15,0 SAY "Are you sure you want to delete"
@ 15,34 SAY NAMES PICTURE "@A"
@ 16,0 SAY "(Y/N)" GET CHECK PICTURE "Y"
READ
*** if they want to delete then pack the database
IF CHECK
  PACK
  CLEAR
  @ 20,0 SAY "Employee was deleted."
  WAIT "Press any key to continue."
*** otherwise do not delete
ELSE
  RECALL
CLEAR
@ 20,0 SAY "Employee was not deleted."
WAIT "Press any key to continue."
ENDIF
ENDIF
RETURN

****** ATTENDAN.PRG
****** ATTENDANCE PROGRAM
****** UNIVERSITY OF MICHIGAN HOSPITAL LAUNDRY SERVICE
****** this program allows the user to enter the attendance
****** for all the employees
****** it accesses atten2.prg throughout
GO TOP
DO WHILE .NOT. EOF()
****** print the heading on the top of the screen
CLEAR
@1,5 SAY 'ENTER TODAY'S EMPLOYEE ATTENDANCE'
@2,5 SAY 'Hit true <T> or <enter> if the employee is present.'
@3,5 SAY 'Hit false <F> if the employee is absent. The computer may beep.'
@4,15 SAY 'NAME | PRESENT?
@5,12 SAY '-------------------------------
DO WHILE .T.
****** this part just prints the name and current attendance
****** of all the employees on the database
@6,15 SAY NAMES+ | '  
@6,48 SAY PRESENT PICTURE '@Y'
SKIP
****** once the end of the file is reached the program must
****** jump back to the first person on the screen
****** this was tricky to do. all of the skip-X commands
****** correct employee
IF EOF()
  SKIP -1
EXIT
ENDIF
@7,15 SAY NAMES+ | '  
@7,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
  SKIP -2
EXIT
ENDIF
@8,15 SAY NAMES+ | '  
@8,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
  SKIP -3
EXIT
ENDIF
@9,15 SAY NAMES+ | '  
@9,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
  SKIP -4
EXIT
ENDIF
@10,15 SAY NAMES+ | '  
@10,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
  SKIP -5
EXIT
ENDIF
@11,15 SAY NAMES+ | '  

GO TOP
DO WHILE .NOT. EOF()
@11,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -6
   EXIT
ENDIF
@12,15 SAY NAMES+ ' |
@12,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -7
   EXIT
ENDIF
@13,15 SAY NAMES+ ' |
@13,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -8
   EXIT
ENDIF
@14,15 SAY NAMES+ ' |
@14,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -9
   EXIT
ENDIF
@15,15 SAY NAMES+ ' |
@15,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -10
   EXIT
ENDIF
@16,15 SAY NAMES+ ' |
@16,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -11
   EXIT
ENDIF
@17,15 SAY NAMES+ ' |
@17,48 SAY PRESENT PICTURE '@Y'
SKIP
IF EOF()
   SKIP -12
   EXIT
ENDIF
@18,15 SAY NAMES+ ' |
@18,48 SAY PRESENT PICTURE '@Y'
SKIP
*** at this point the end of the screen is reached and the program
*** must jump back to the employee at the top of the screen
IF EOF()
   SKIP -13
   EXIT
ENDIF
@19,15 SAY NAMES+ ' |
@19,48 SAY PRESENT PICTURE '@Y'
SKIP -13
EXIT
ENDDO
*** atten2 allows the user to enter the attendance
DO ATTN2
ENDDO
RETURN
***** SCHPROG.PRG
***** SCHEDULING PROGRAM
***** UMH LAUNDRY SERVICES
***** this program begins the daily scheduling
***** this index alphabetizes the database so that
***** the attendance can be entered easier
INDEX ON NAMES TO NAMEINDX
SET INDEX TO NAMEINDX
DO WHILE .T.
**** this section remembers the day entered in laundry.prg
**** and then calls the appropriate sub program
**** so that the correct number of jobs at each station
**** will be entered for each day
IF DAY='MO' .OR. DAY='TU' .OR. DAY='WE' .OR. DAY='TH'
  DO JOBNUM1
ENDIF
IF DAY='FR' .OR. DAY='SA'
  DO JOBNUM2
ENDIF
IF DAY='SP'
  DO SPECIAL
ENDIF
CLEAR
**** this program does the attendance
DO ATTENDAN
CLEAR
@1,1 TO 19,79
@8,27 SAY 'ATTENDANCE IS COMPLETED'
**** to return to the attendance menu hit any key
**** if a C is hit the computer will continue with the
**** scheduling program
@11,10 TO 14,69 DOUBLE
@12,12 say 'Hit a <C> to continue with the scheduling program.'
@13,12 say 'Hit any key to return to the employee attendance section.'
WAIT '' TO CHOICE
IF UPPER(CHOICE)='C'
**** if C is chosen the computer moves on
**** first the index is closed and deleted so that
**** it may be re-indexed later
  CLOSE INDEX
  DELETE FILE NAMEINDX.NDX
  EXIT
**** exiting makes the program leave the do loop
**** otherwise the program returns to attendance program
ENDIF
ENDDO
CLEAR
@9,20 SAY 'The computer is sorting the jobs.'
@10,20 say 'This will take approximately 5 minutes.'
@8,15 TO 11,64 DOUBLE
**** this section calls the necessary subprograms
**** to complete the scheduling
**** classify.prg sorts the workers into their own
**** job classification
DO CLASSIFY
**** tempjobs sorts the people into their temporary
**** promotions for the day
DO TEMPJOBS
**** printout simply prints the listing of the
**** workers jobs for the day as worked out
**** by the program
DO PRINTOUT
RETURN
****** TEMPJOBS.PRG
****** THIS PROGRAM MOVES PEOPLE TO THEIR TEMPORARY JOBS
****** UNIVERSITY OF MICHIGAN HOSPITALS
GO TOP
DO WHILE .T.
    **** as soon as we reach the end leave this program
    IF EOF()
        EXIT
    ENDF
    **** if they are already in their favorite job then
    **** go to the next person
    IF SCORE=1
        SKIP
        LOOP
    ENDF
    **** if the person is not here then go on to the next
    **** person
    IF .NOT. PRESENT
        SKIP
        LOOP
    ENDF
    STORE SCORE TO TEMPScore
    **** this section compares each job with the employees
    **** preferences and the temporary job openings. I will
    **** use the first one as an example
    **** if the worker prefers to work on the primary iron
    **** more than their current job (ffprimary<score)
    **** and the preference of ffprimary does not equal
    **** zero meaning that it is not preferred (ffprimary<>0)
    **** and there is an opening for a temporary promotion
    **** (nffprimary>0) then place the employee in that job
    IF FFPREIMARY<SCORE .AND. FFPREIMARY>0 .AND. NFFPRIMARY>0
    **** job sort moves them out of their old job by adding
    **** one to the job classification number variable
    **** hence 'n'+'classific'+1
    DO JOBSORT
        **** this changes their score to the preference of their
        **** new temporary job
        REPLACE SCORE WITH FFPREIMARY
        **** this removes one job space from the primary iron
        **** this is done by subtracting one from the classification
        **** number variable
        STORE NFFPRIMARY-1 TO NFFPRIMARY
    ENDF
    **** the following are similar to above
    IF FFNUMTHREE<SCORE .AND. FFNUMTHREE>0 .AND. NFFNUMTHREE>0
        DO JOBSORT
            REPLACE SCORE WITH FFNUMTHREE
            STORE NFFNUMTHREE-1 TO NFFNUMTHREE
        ENDF
    IF FFSWING<SCORE .AND. FFSWING>0 .AND. NFFSWING>0
        DO JOBSORT
            REPLACE SCORE WITH FFSWING
            STORE NFFSWING-1 TO NFFSWING
        ENDF
    IF FFMACHFOLD<SCORE .AND. FFMACHFOLD>0 .AND. NFFMACHFOLD>0
        DO JOBSORT
            REPLACE SCORE WITH FFMACHFOLD
            STORE NFFMACHFOLD-1 TO NFFMACHFOLD
        ENDF
    IF FFMISC<SCORE .AND. FFMISC>0 .AND. NFFMISC>0
        DO JOBSORT
            REPLACE SCORE WITH FFMISC
            STORE NFFMISC-1 TO NFFMISC
        ENDF
    IF CLSOILSORT<SCORE .AND. CLSOILSORT>0 .AND. NCLSOILSORT>0
        DO JOBSORT
            REPLACE SCORE WITH CLSOILSORT
            STORE NCLSOILSORT-1 TO NCLSOILSORT
ENDIF
IF CLCAMPUS<SCORE .AND. CLCAMPUS>0 .AND. NCLCAMPUS>0
DO JOBSORT
REPLACE SCORE WITH CLCAMPUS
STORE NCLCAMPUS-1 TO NCLCAMPUS
ENDIF
IF CLBASKET<SCORE .AND. CLBASKET>0 .AND. NCLBASKET>0
DO JOBSORT
REPLACE SCORE WITH CLBASKET
STORE NCLBASKET-1 TO NCLBASKET
ENDIF
IF CLFLOOR<SCORE .AND. CLFLOOR>0 .AND. NCLFLOOR>0
DO JOBSORT
REPLACE SCORE WITH CLFLOOR
STORE NCLFLOOR-1 TO NCLFLOOR
ENDIF
IF CUSTODIAN<SCORE .AND. CUSTODIAN>0 .AND. NCUSTODIAN>0
DO JOBSORT
REPLACE SCORE WITH CUSTODIAN
STORE NCUSTODIAN-1 TO NCUSTODIAN
ENDIF
IF STOCKKEEP<SCORE .AND. STOCKKEEP>0 .AND. NSTOCKKEEP>0
DO JOBSORT
REPLACE SCORE WITH STOCKKEEP
STORE NSTOCKKEEP-1 TO NSTOCKKEEP
ENDIF
IF CHECKHOSP<SCORE .AND. CHECKHOSP>0 .AND. NCHECKHOSP>0
DO JOBSORT
REPLACE SCORE WITH CHECKHOSP
STORE NCHECKHOSP-1 TO NCHECKHOSP
ENDIF
IF CHECKTRAYS<SCORE .AND. CHECKTRAYS>0 .AND. NCHECKTRAYS>0
DO JOBSORT
REPLACE SCORE WITH CHECKTRAYS
STORE NCHECKTRAYS-1 TO NCHECKTRAYS
ENDIF
IF CHECKCAMPU<SCORE .AND. CHECKCAMPU>0 .AND. NCHECKCAMPU>0
DO JOBSORT
REPLACE SCORE WITH CHECKCAMPU
STORE NCHECKCAMPU-1 TO NCHECKCAMPU
ENDIF
IF WL<SCORE .AND. WL>0 .AND. NWL>0
DO JOBSORT
REPLACE SCORE WITH WL
STORE NWL-1 TO NWL
ENDIF
IF OLM<SCORE .AND. OLM>0 .AND. NOLM>0
DO JOBSORT
REPLACE SCORE WITH OLM
STORE NOLM-1 TO NOLM
ENDIF
IF OMV<SCORE .AND. OMV>0 .AND. NOMV>0
DO JOBSORT
REPLACE SCORE WITH OMV
STORE NOMV-1 TO NOMV
ENDIF
IF MACHREP<SCORE .AND. MACHREP>0 .AND. NMACHREP>0
DO JOBSORT
REPLACE SCORE WITH MACHREP
STORE NMACHREP-1 TO NMACHREP
ENDIF
IF GARMENT<SCORE .AND. GARMENT>0 .AND. NGARMENT>0
DO JOBSORT
REPLACE SCORE WITH GARMENT
STORE NGARMENT-1 TO NGARMENT
ENDIF
IF PRESSOP<SCORE .AND. PRESSOP>0 .AND. NPRESSOP>0
    DO JOBSORT
    REPLACE SCORE WITH PRESSOP
    STORE NPRESSOP-1 TO NPRESSOP
ENDIF

if the computer has not found a new temporary job then
**** move on to the next person
IF TEMPSCORE=SCORE
    SKIP
**** if the computer did find a new job then start this whole
**** program over. this ensures that the people with the
**** highest seniority get their favorite jobs
**** as per the union contract
ELSE
    GO TOP
ENDIF
ENDDO
RETURN

**** newemply.prg
**** this program enters a new employee
**** U OF M HOSPITALS
CLEAR
SET TALK OFF
USE SCHEDULE
GO BOTTOM
**** set the screen to the format insmen
SET FORMAT TO INS MEN
INSERT
**** now the employee has been entered
CLOSE FORMAT
CLEAR
RETURN

**** scheduling program, university of michigan laundry services
**** schedule.prg
**** this program simply provides for the main menu
CLEAR
STORE '' TO CHOICE
**** begin the menu
DO WHILE .T.
**** print choices
CLEAR
@1,0 TO 19,79
@2,10 SAY '**** main menu for umh laundry services scheduling program ****'
@3,1 TO 3,78
@17,1 TO 17,78
@6,24 SAY '<Q>....QUIT [RETURN TO DOT PROMPT]'
@7,24 SAY '<S>....SCHEDULING FOR THE DAY'
@8,24 SAY '<P>....PREFERENCES CHANGE'
@9,24 SAY '<A>....ADD AN EMPLOYEE'
@10,24 SAY '<D>....DELETE AN EMPLOYEE'
@11,24 SAY '<L>....LISTING OF DATABASE'
@12,24 SAY '<C>....CHANGE PERMANENT POSITIONS'
@13,24 SAY '<F>....FORM FOR PREFERENCE CHANGES'
@18,31 SAY 'ENTER CHOICE'
**** enter the choice to the variable choice
WAIT '"' TO CHOICE
**** the case loop checks to see which choice was entered
DO CASE
    CASE UPPER(CHOICE)='Q'
        **** if quitting close the database and return
        CLOSE DATABASES
        CLEAR
        RETURN
    CASE UPPER(CHOICE)='S'
        **** schprog is the scheduling program
        DO SCHPROG
    CASE UPPER(CHOICE)='A'

**** addemply is the program to add a new employee
   DO ADDEMPLY
   CASE UPPER(CHOICE)='D'
**** delemply removes an employee from the database
   DO DELEMPLY
   CASE UPPER(CHOICE)='P'
*** prefer changes one employees preferences
   DO PREFER
   CASE UPPER(CHOICE)='I'
*** eprint provides a menu for printing the database
   DO EPRINT
   CASE UPPER(CHOICE)='C'
*** permjob changes the permanant classification of an
*** employee
   DO PERMJOB
   CASE UPPER(CHOICE)='F'
*** form print a preferences change form
   DO FORM
   OTHERWISE
*** if something incorrect was entered for choice
*** it returns to the top and starts over
   LOOP
ENDCASE
ENDDO

********** JOBSORT.PRG
********** THIS PROGRAM HELPS SORT EMPLOYEES INTO THIER TEMP JOBS
********** U OF M HOSPITALS
*** all this program does is to check to see what the
*** job is that they are currently assigned
*** this occurs just before they are transferred to a new
*** job. It then adds one to the job classification number
*** which indicates that there is now one more job
*** available for that specific work area
   DO CASE
   ***example
   *** if your current job is feeder folder primary iron
   CASE SCORE=FFPRIMARY
   *** then let the computer know there is now one more
   *** job available at this location
   STORE NFFPRIMARY+1 TO NFFPRIMARY
   *** now return to the program that makes the employee
   *** leave this job and gets another
   RETURN
   CASE SCORE=FFNUMTHREE
   STORE NFFNUMTHREE+1 TO NFFNUMTHREE
   RETURN
   CASE SCORE=FFSWING
   STORE NFFSWING+1 TO NFFSWING
   RETURN
   CASE SCORE=FFMACHFOLD
   STORE NFMACHFOLD+1 TO NFMACHFOLD
   RETURN
   CASE SCORE=FFMISC
   STORE NFMISC+1 TO NFMISC
   RETURN
   CASE SCORE=CLSOILSORT
   STORE NCLSOILSORT+1 TO NCLSOILSORT
   RETURN
   CASE SCORE=CLCAMPUS
   STORE NCLCAMPUS+1 TO NCLCAMPUS
   RETURN
   CASE SCORE=CLBASKET
   STORE NCLBASKET+1 TO NCLBASKET
   RETURN
   CASE SCORE=CLFLOOR
@10,20 SAY 'Is this the employee you want? (Y/N)'
@12,30 SAY NAMES
@9,15 to 13,64 double
WAIT '' TO CHOICE
IF UPPER(CHOICE)='N'
    CONTINUE
ENDIF
ENDDO
**** if the employee was not found prompt the user
IF EOF()
    @10,20 SAY 'Employee not found. Please check spelling'
    @18,30 SAY 'Hit any key to continue.'
    WAIT'' TO WAITS
    RETURN
ENDIF
*** the program does not get this far unless the employee
*** was found, and the user entered y you have found the
*** right employee
DO WHILE .T.
    CLEAR
    @1,5 SAY NAMES+ ' Current Classification: '+CLASSIFIC
    @2,1 to 2,79
    *** show a listing of the classifications
    @4,10 SAY 'A) Feeder Folder Primary'
    B) Feeder Folder Number Three'
    @5,10 say 'C) Feeder Folder Machine Fold'
    D) Feeder Folder Miscellaneous'
    @6,10 say 'E) Classifier Soil Sort'
    F) Classifier Campus'
    @7,10 say 'G) Classifier Basket'
    H) Classifier Floor'
    @8,10 say 'I) Custodian'
    J) Stockkeeper'
    @9,10 say 'K) Checker Hospital'
    L) Checker Trays'
    @10,10 say 'M) Checker Campus'
    N) Weigher Loader'
    @11,10 say 'O) Operator Laundry Machine'
    P) Operator Motor Vehicle'
    @12,10 say 'Q) Machine Repair'
    R) Garment'
    @13,10 say 'S) Press Operator'
    T) Feeder Folder Swing Crew'
    @18,12 say 'Enter the letter of the employees new job classification.'
    @17,10 TO 19,69 DOUBLE
    *** ask the user to enter the code letter of the new
    *** job classification to variable choice
    WAIT '' TO CHOICE
    *** this CASE loop checks to see which letter was entered
    *** it then stores the classification to tempname
    *** if verified later tempname will be stored
    *** to CLASSIFIC of the employee
    DO CASE
        CASE UPPER(CHOICE)='A'
            STORE 'FFPRIMARY' TO TEMPNAME
        CASE UPPER(CHOICE)='B'
            STORE 'FFNUMTHREE' TO TEMPNAME
        CASE UPPER(CHOICE)='C'
            STORE 'FFMACHFOLD' TO TEMPNAME
        CASE UPPER(CHOICE)='D'
            STORE 'FFMISC' TO TEMPNAME
        CASE UPPER(CHOICE)='E'
            STORE 'CLSOILSORT' TO TEMPNAME
        CASE UPPER(CHOICE)='F'
            STORE 'CLCAMPUS' TO TEMPNAME
        CASE UPPER(CHOICE)='G'
            STORE 'CLBASKET' TO TEMPNAME
        CASE UPPER(CHOICE)='H'
            STORE 'CLFLOOR' TO TEMPNAME
        CASE UPPER(CHOICE)='I'
            STORE 'CUSTODIAN' TO TEMPNAME
        CASE UPPER(CHOICE)='J'
            STORE 'STOCKKEEP' TO TEMPNAME
        CASE UPPER(CHOICE)='K'
            STORE 'CHECKHOSP' TO TEMPNAME
STORE NCLFLOOR+1 TO NCLFLOOR
RETURN
CASE SCORE=CUSTODIAN
    STORE NCUSTOMIAN+1 TO NCUSTOMIAN
RETURN
CASE SCORE=STOCKKEEP
    STORE NSTOCKKEEP+1 TO NSTOCKKEEP
RETURN
CASE SCORE=CHECKHOSP
    STORE NCHECKHOSP+1 TO NCHECKHOSP
RETURN
CASE SCORE=CHECKTRAYS
    STORE NCHECKTRAYS+1 TO NCHECKTRAYS
RETURN
CASE SCORE=CHECKCAMPU
    STORE NCHECKCAMPU+1 TO NCHECKCAMPU
RETURN
CASE SCORE=WL
    STORE NWL+1 TO NWL
RETURN
CASE SCORE=OLM
    STORE NOLM+1 TO NOLM
RETURN
CASE SCORE=OMV
    STORE NOMV+1 TO NOMV
RETURN
CASE SCORE=MACHREP
    STORE NMACHREP+1 TO NMACHREP
RETURN
CASE SCORE= GARMENT
    STORE NGARMENT+1 TO NGARMENT
RETURN
CASE SCORE=PRESSOP
    STORE NPRESSOP+1 TO PRESSOP
RETURN
OTHERWISE
    RETURN
ENDCASE
***** permjob.prg
***** changes the permanent job classification of employees
***** University of Michigan Hospitals
*****
CLEAR
**** tell user what program he is in
@10,10 SAY 'This program will change the permanent classification'
@11,10 SAY 'of a given employee.'
@18,10 SAY 'Hit any key to continue. Hit a <Q> to quit.'
@9,5 TO 19,74 double
**** if they want to exit this program the user enters
*** q to choice
   wait '' TO CHOICE
   IF UPPER(CHOICE)='Q'
      RETURN
ENDIF
STORE 'N' TO CHOICE
*** prompt the user to enter the last name of the
*** employee to be found
*** enter the name to tempname
CLEAR
@10,1 SAY ''
ACCEPT 'Enter the last name of the employee you wish to find. ' TO TEMPNAME
LOCATE FOR UPPER(NAMES)=UPPER(TEMPNAME)
*** look for the employee. keep looking until the user
*** enters y-- you have found the correct employee
DO WHILE UPPER(CHOICE)<>'Y' .AND. .NOT. EOF()
    CLEAR
CASE UPPER(CHOICE)='L'
  STORE 'CHECKTRAYS' TO TEMPNAME
CASE UPPER(CHOICE)='M'
  STORE 'CHECKCAMPU' TO TEMPNAME
CASE UPPER(CHOICE)='N'
  STORE 'WEL' TO TEMPNAME
CASE UPPER(CHOICE)='O'
  STORE 'OLM' TO TEMPNAME
CASE UPPER(CHOICE)='P'
  STORE 'OMV' TO TEMPNAME
CASE UPPER(CHOICE)='Q'
  STORE 'MACHREP' TO TEMPNAME
CASE UPPER(CHOICE)='R'
  STORE 'GARMENT' TO TEMPNAME
CASE UPPER(CHOICE)='S'
  STORE 'PRESSOP' TO TEMPNAME
CASE UPPER(CHOICE)='T'
  STORE 'FFSWING' TO TEMPNAME
OTHERWISE
  LOOP
ENDCASE
CLEAR

*** this section verifies the choice of the user
@5,35 SAY 'IMPORTANT!!'
@9,10 SAY 'Is '+LTRIM(RTRIM(NAMES))+"'s permanent job to be changed from'
@11,25 SAY LTRIM(RTRIM(CLASSIFIC))+" to '+TEMPNAME+'?'
@15,35 SAY 'IMPORTANT!!'
@8,5 TO 12,74 DOUBLE
@18,28 SAY 'Enter (Y/N) or <Q> to quit'
WAIT '' TO WAITS

*** if q is entered the program exits
IF UPPER(WAITS)='Q'
  EXIT
ENDIF

*** if y is chosen the program replaces CLASSIFIC with TEMPNAME
*** in order to store the new classification to the database
IF UPPER(WAITS)='Y'
  REPLACE CLASSIFIC WITH TEMPNAME
  CLEAR
  @10,10 SAY 'DATABASE UPDATED! Hit any key to change preferences.'
  WAIT '' TO WAITS
  SET FORMAT TO PREFEREN
  EDIT NEXT 1
  CLOSE FORMAT

**** if n was entered for WAITS then the program
**** returns to the top to start over
RETURN
ENDDO
ENDDO
RETURN

****** PRINTS OUT A FORM FOR PREFERENCES CHANGES
****** UNIVERSITY OF MICHIGAN HOSPITALS
****** all this program does is generate a
****** preference change form
CLEAR
@10,34 SAY 'PREFERENCES FORM'
@15,23 SAY 'Ready printer. Hit any key to print.'
WAIT '' TO WAITS
ET PRINT ON
' '
' '
' '
' '
' '
' '
' '
'

PREFERENCES CHANGE FORM'
Make sure that you only rank those jobs that you are qualified for.
Order your preferences from favorite to least favorite. (1,2,3....)

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeder Folder Primary</td>
</tr>
<tr>
<td>Feeder Folder Number Three</td>
</tr>
<tr>
<td>Feeder Folder Swing Crew</td>
</tr>
<tr>
<td>Feeder Folder Machine Fold</td>
</tr>
<tr>
<td>Feeder Folder Miscellaneous</td>
</tr>
<tr>
<td>Classifier Soil Sort</td>
</tr>
<tr>
<td>Classifier Campus</td>
</tr>
<tr>
<td>Classifier Basket</td>
</tr>
<tr>
<td>Classifier Floor</td>
</tr>
<tr>
<td>Custodian</td>
</tr>
<tr>
<td>Stockkeeper</td>
</tr>
<tr>
<td>Checker Hospital</td>
</tr>
<tr>
<td>Checker Trays</td>
</tr>
<tr>
<td>Checker Campus</td>
</tr>
<tr>
<td>Weigher Loader</td>
</tr>
<tr>
<td>OLM (Operator Laundry Machine)</td>
</tr>
<tr>
<td>OMV (Operator Motor Vehicle)</td>
</tr>
<tr>
<td>Machine Repair</td>
</tr>
<tr>
<td>Garment</td>
</tr>
<tr>
<td>Press Operator</td>
</tr>
</tbody>
</table>

set print off
eject
return
**** prefer.prg
**** this program changes the preferences of the employees
**** u of m hospitals
CLEAR
SE SCHEDULE
SET TALK OFF
SET SCOREBOARD OFF
SET STATUS OFF
*** initialize verf
STORE 'n' TO VERF
CHECK=/.T.
**** create a menu to ask the employees last name
@ 4, 7 TO 14, 64
@ 7, 25 SAY "PREFERENCE CHANGE OPTION"
@ 9, 15 SAY "PLEASE ENTER THE LAST NAME OF THE EMPLOYEE"
@ 10, 15 SAY "WHOSE PREFERENCES YOU WOULD LIKE TO CHANGE."

**** enter the last name into tempo
ACCEPT ' ' TO TEMPO

LOCATE FOR UPPER(NAMES)=UPPER(TEMPO)

**** if the computer finds the right person then
**** the user enters y and the program continues
DO WHILE UPPER(VERF)<>‘Y’ .AND. .NOT. EOF()
  CLEAR

**** prompt for a yes no answer
@ 10,20 SAY 'Is the employee you want to change? (Y/N)'
@ 12,30 SAY "GET VERF PICTURE "y"
READ

*** store the yes no answer to verf
@ 14,30 SAY " " GET VERF PICTURE "y"

*** if this is not the employee then look for
*** someone else with the same spelling
  IF UPPER(VERF)=‘N’
  CONTINUE
  ENDIF
ENDDO

*** if the computer didn't find anyone it gives this message
IF EOF()
  CLEAR
  @ 10,20 SAY "Employee not found. Please check spelling."
  @ 11,20 SAY "No other employee by that name was found."
  @ 13,20 SAY "Hit any key to continue."
  WAIT ' ' TO WAITS
RETURN
ELSE

*** the program only gets this far if the computer
*** found the right employee and the user entered
*** a yes into verf
  CLEAR

*** the format reads in the preferences
  SET FORMAT TO PREFEREN
  EDIT NEXT 1
  CLOSE FORMAT
ENDIF

RETURN

***** printout.prg

****** PRINTS OUT A LISTING OF THE SCHEDULED JOBS
****** UNIVERSITY OF MICHIGAN HOSPITALS

***** this program generates a printout from the scheduling
*** program
CLEAR

*** prompt the user to ready the printer
@10,35 SAY 'SORTING COMPLETE'
@15,25 SAY 'Ready printer. Hit any key to print.'
WAIT ' ' TO WAITS

STORE 1 TO NUM
SET PRINT ON

**** this part prints the classifications
DO WHILE .T.

*** NUM is a variable used to make sure that each section
*** of the following program is only accessed once
** example
** if the variable num=1
IF NUM=1

*** find all the people that are now assigned
*** to the primary iron
*** note if they are assigned to the primary iron
*** their score equals the ranking they gave for
*** the primary iron
-*** locate causes the computer
-*** to find the first person in the database
-*** with the primary iron job
-LOCATE FOR FFPRIMARY=SCORE
-*** sub heading for the listing
-?'Feeder Folders Primary Iron'
-*** if not all the job spaces for the primary iron
-*** were filled then show this on the
-*** printout
-IF NFPFRIMARY>0
-*** this says X number of jobs were not filled
-*** for the primary iron
-? STR(NFPFRIMARY)' EMPTY JOB SPACE DUE TO ABSENCES'
-*** because num=1 for this pass all of the following
-*** if statements will be skipped until the very end
-*** jump now to the end
-ENDIF
-ENDIF
-IF NUM=2
-LOCATE FOR FFSNUMTHREE=SCORE
-?'Feeder Folders Number Three Iron'
-IF NFFNUMTHREE>0
-? STR(NFFNUMTHREE)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=3
-LOCATE FOR FFSWING=SCORE
-?'Feeder Folders Swing Crew ****'
-IF NFFSWING>0
-? STR(NFFSWING)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=4
-LOCATE FOR FFMACHFOLD=SCORE
-?'Feeder Folder Machine Fold'
-IF NFFMACHFOLD>0
-? STR(NFFMACHFOLD)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=5
-LOCATE FOR FFMISC=SCORE
-?'Feeder Folder Miscellaneous'
-IF NFFMISC>0
-? STR(NFFMISC)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=6
-LOCATE FOR CLSOILSORT=SCORE
-?'Classifiers Soil Sort'
-IF NCLSOILSORT>0
-? STR(NCLSOILSORT)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=7
-LOCATE FOR CLCAMPUS=SCORE
-?'Classifiers Campus'
-IF NCLCAMPUS>0
-? STR(NCLCAMPUS)' EMPTY JOB SPACE DUE TO ABSENCES'
-ENDIF
-ENDIF
-IF NUM=8
-LOCATE FOR CLBASKET=SCORE
-?'Classifiers Basket'
-IF NCLBASKET>0
-? STR(NCLBASKET)' EMPTY JOB SPACE DUE TO ABSENCES'
IF NUM=9
LOCATE FOR CLFLOOR=SCORE
?'Classifiers Floor'
IF NCLBASKET>0
  ? STR(NCLFLOOR)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=10
LOCATE FOR CUSTODIAN=SCORE
?'Custodian'
IF NCUSTODIAN>0
  ? STR(NCUSTODIAN)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=11
LOCATE FOR STOCKKEEP=SCORE
?'Stockkeeper'
IF NSTOCKKEEP>0
  ? STR(NSTOCKKEEP)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=12
LOCATE FOR CHECKHOSP=SCORE
?'Checkers Hospital'
IF NCHECKHOSP>0
  ? STR(NCHECKHOSP)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=13
LOCATE FOR CHECKTRAYS=SCORE
?'Checkers Trays'
IF NCHECKTRAYS>0
  ? STR(NCHECKTRAYS)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=14
LOCATE FOR CHECKCAMPUS=SCORE
?'Checkers Campus'
IF NCHECKCAMPUS>0
  ? STR(NCHECKCAMPUS)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=15
LOCATE FOR WL=SCORE
?'Weigher Loader'
IF NWL>0
  ? STR(NWL)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=16
LOCATE FOR OLM=SCORE
?'Operator Laundry Machine'
IF NOLM>0
  ? STR(NOLM)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=17
LOCATE FOR OMV=SCORE
?'Operator Motor Vehicle'
IF NOMV>0
  ? STR(NOMV)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF
IF NUM=18
LOCATE FOR MACHREP=SCORE
?'Machine Repair'
IF NMACHREP>0
   ? STR(NOMV)+' EMPTY JOB SPACE DUE TO ABSENCES'
ENDIF
ENDIF

IF NUM=19
   LOCATE FOR GARMENT=SCORE
   ?'Garments'
   IF NGARMENT>0
      ? STR(NGARMENT)+' EMPTY JOB SPACE DUE TO ABSENCES'
   ENDIF
ENDIF

IF NUM=20
   LOCATE FOR PRESSOP=SCORE
   ?'Press Operator'
   IF NPRESSOP>0
      ? STR(NPRESSOP)+' EMPTY JOB SPACE DUE TO ABSENCES'
   ENDIF
ENDIF

IF NUM=21
   LOCATE FOR PRESENT .AND. SCORE=99
   ?'These people were not assigned jobs.'
ENDIF

IF NUM=22
   LOCATE FOR .NOT. PRESENT
   ?'Absent today'
ENDIF

IF NUM=23
   EXIT
ENDIF

*** now add one to num so that the same heading
*** will not be entered twice
STORE NUM+1 TO NUM

*** now print all of the people that will have
*** the job as indicated in the above section
*** it is looking for the classification
*** as appeared in the LOCATE FOR _________
DO WHILE .NOT. EOF()
   ?' ' +NAMES+ ' ' +classific+'___'
   continue ??'
      ?' '+names+' ' +classific+'___'
ENDDO

**** continue causes the computer to look for the
**** next person on the database with the specifications
**** as indicated in the LOCATE statement
CONTINUE

ENDDO
?
?
?

ENDDO

?'
?
?

SET PRINT OFF
EJECT
RETURN