

|  |  |
| --- | --- |
| student name: | MUKABAK ORAZBEK |
| intake code: |
| module name: |
| DUE DATE: |
| lecturer: |

**Table of contents**

[Introduction 3](#_Toc489341271)

[Header Files 4](#_Toc489341272)

[Global Variables 5](#_Toc489341273)

[Structs 6](#_Toc489341274)

[Functions 8](#_Toc489341275)

[Main Function 8](#_Toc489341276)

[Main Menu 8](#_Toc489341277)

[Admin Menu 9](#_Toc489341278)

[Lecturer Menu 10](#_Toc489341279)

[Student Menu 12](#_Toc489341280)

[Attendance 13](#_Toc489341281)

[Take Attendance 13](#_Toc489341282)

[Modify Attendance 17](#_Toc489341283)

[View My Attendance 21](#_Toc489341284)

[Users 23](#_Toc489341285)

[Add Lecturer & Student 23](#_Toc489341286)

[Search Lecturer & Student 24](#_Toc489341287)

[Modify Lecturer & Student 27](#_Toc489341288)

[Delete Lecturer & Student 29](#_Toc489341289)

[View All Lecturer & Students 31](#_Toc489341290)

[conclusion 32](#_Toc489341291)

[references 33](#_Toc489341292)

# Introduction

# Header Files

INCLUDE stdio h

INCLUDE stdlib h

INCLUDE string h

INCLUDE time h

INCLUDE stdlib h

# Global Variables

SET STRING chUsername LENGTH [16]

SET STRING chUserPassword LENGTH [16]

SET INT intUserType

SET STRING chAdminUsername LENGTH [16] VALUE "abdullah"

SET STRING chAdminPassword LENGTH [16] VALUE "123"

SET STRING dbLoginList VALUE "dbLoginList txt"

SET STRING dbIntakeList VALUE "dbIntakeList txt"

SET STRING dbModuleList VALUE "dbModuleList txt"

SET STRING dbIntakeRecord VALUE "dbIntakeRecord txt"

SET STRING dbModuleRecord VALUE "dbModuleRecord txt"

SET STRING dbAttendance VALUE "dbAttendance txt"

# Structs

STRUCT stLogin

SET INT intType

SET STRING chTP LENGTH [16]

SET STRING chPass LENGTH [16]

SET STRING chName LENGTH [30]

END

STRUCT stIntakeList

SET STRING chIntakeCode LENGTH [16]

END

STRUCT stModuleList

SET STRING chModuleName LENGTH [16]

END

STRUCT stIntakeRecord

SET STRING chIndex LENGTH [16]

SET STRING chUserID LENGTH [16]

SET STRING chIntakeCode LENGTH [16]

END

STRUCT stAttendance

SET STRING chDate LENGTH [16]

SET STRING chStartTime LENGTH [16]

SET STRING chEndTime LENGTH [16]

SET STRING chModuleName LENGTH [16]

SET STRING chStudentName LENGTH [30]

SET STRING chIntakeCode LENGTH [16]

SET STRING chLecturerID LENGTH [16]

SET STRING chStudentTP LENGTH [16]

SET INT intStatus

END

# Functions

## Main Function

Function\_Main ()

CALL Function\_Main()

RETURN

END

## Main Menu

Function\_vMainMenu()

SET STRING chChoice

DOWHILE(1)

CLEAR SCREEN

DISPLAY ("ATTENDANCE MANAGEMENT SYSTEM ")

DISPLAY ("1 ADMIN")

DISPLAY ("2 LECTURER")

DISPLAY ("3 STUDENT")

DISPLAY ("0 EXIT")

DISPLAY ("Please Enter Your Choice: ")

READ (Type\_Chacacter, chChoice)

CLEAR BUFFERING

SWITCH(chChoice) THEN

CASE '1':

CALL vCallAdmin()

BREAK

CASE '2':

CALL vCallLecturer()

BREAK

CASE '3':

CALL vCallStudent()

BREAK

CASE '0':

exit(0)

DEFAULT:

CALL Function\_Main()

BREAK

END

CALL vGet()

END

RETURN

END

### Admin Menu

Function\_vAdminMenu()

SET STRING chChoice

DOWHILE(1)

CLEAR SCREEN

DISPLAY ("ADMIN MENU ")

DISPLAY ("1 MANAGE STUDENT")

DISPLAY ("2 MANAGE LECTURE")

DISPLAY ("3 MANAGE INTAKE")

DISPLAY ("4 MANAGE MODULE")

DISPLAY ("5 MODIFY ATTENDANCE")

DISPLAY ("0 LOG OUT")

DISPLAY ("")

DISPLAY ("Please Enter Your Choice: ")

READ (Type\_Chacacter, chChoice)

CLEAR BUFFERING

SWITCH(chChoice) THEN

CASE '1':

CALL vManageStudentMenu()

BREAK

CASE '2':

CALL vManageLecturerMenu()

BREAK

CASE '3':

CALL vManageIntakeMenu()

BREAK

CASE '4':

CALL vManageModuleMenu()

BREAK

CASE '5':

CALL vAdminModIFyAttendance()

BREAK

CASE '0':

CALL main()

DEFAULT:

CALL vAdminMenu()

BREAK

END

CALL vGet()

END

RETURN

END

### Lecturer Menu

Function\_vLecturerMenu()

SET STRING chChoice

DOWHILE(1)

CLEAR SCREEN

DISPLAY ( LECTURER MENU")

DISPLAY ("1 TAKE ATTENDANCE")

DISPLAY ("2 MODIFY ATTENDANCE")

DISPLAY ("3 VIEW ALL STUDENTS")

DISPLAY ("4 VIEW MY INTAKES")

DISPLAY ("5 VIEW MY MODULES")

DISPLAY ("0 LOG OUT")

DISPLAY ("")

DISPLAY ("Please Enter Your Choice: ")

READ (Type\_Chacacter, chChoice)

CLEAR BUFFERING

SWITCH(chChoice) THEN

CASE '1':

CALL vTakeAttendance()

BREAK

CASE '2':

CALL vModIFyAttendance()

BREAK

CASE '3':

CALL vViewAllStudents()

BREAK

CASE '4':

CALL vViewMyIntakes()

BREAK

CASE '5':

CALL vViewMyModules()

BREAK

CASE '0':

CALL main()

DEFAULT:

vLecturerMenu()

BREAK

END

CALL vGet()

END

RETURN

END

### Student Menu

Function\_vStudentMenu()

SET STRING chChoice

DOWHILE(1)

CLEAR SCREEN

DISPLAY ( STUDENT MENU ==")

DISPLAY ("1 VIEW MY ATTENDANCE")

DISPLAY ("2 VIEW MY MODULES")

DISPLAY ("0 LOG OUT")

DISPLAY ("")

DISPLAY ("Please Enter Your Choice: ")

READ (Type\_Chacacter, chChoice)

CLEAR BUFFERING

SWITCH(chChoice) THEN

CASE '1':

CALL vViewAttendance()

BREAK

CASE '2':

CALL vViewMyModules()

BREAK

CASE '0':

CALL main()

DEFAULT:

CALL vStudentMenu()

BREAK

END

CALL vGet()

END

RETURN

END

## Attendance

### Take Attendance

Function\_vTakeAttendance()

CLEAR SCREEN

DISPLAY ("TAKE ATTENDANCE")

SET INT intFound1=0, intFound2=0, intCompare

STRUCT stLogin st

STRUCT stIntakeRecord st1

STRUCT stModuleRecord st2

SET FILE \*fp, \*fp1, \*fp2, \*fp3

fp=OPEN FILE(dbLoginList, "READ MODE")

fp1=OPEN FILE(dbIntakeRecord, "READ MODE")

fp2=OPEN FILE(dbModuleRecord, "READ MODE")

fp3=OPEN FILE(dbAttendance, "READ AND WRITE MODE")

stA intStatus=1

GET AND SET (stA chLecturerID, chUsername, 16)

DISPLAY ("List Of Intakes: ", chUsername)

DOWHILE(1)

READ FILE (st1, GET SIZE(st1), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(chUsername, st1 chUserID)

IF(intCompare==0) THEN

DISPLAY ("Type String", st1 chIntakeCode)

intFound1=1

END IF

END DOWHILE

IF(intFound1==0)

DISPLAY ("NOT FOUND!")

END IF

FILE CLOSE(fp1)

DISPLAY ("Please Select The Intake Code: ")

READ ("Type String", stA chIntakeCode)

CALL vUpper(stA chIntakeCode)

CLEAR BUFFERING

DISPLAY ("List Of Modules:", chUsername)

DOWHILE(1)

READ FILE (st2, GET SIZE(st2), 1, fp2)

IF(READING(fp2)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(chUsername, st2 chUserID)

IF(intCompare==0) THEN

DISPLAY ("Type String", st2 chModuleName)

intFound2=1

END IF

END DOWHILE

IF(intFound2==0) THEN

DISPLAY ("NOT FOUND!")

END IF

FILE CLOSE(fp2)

DISPLAY ("Please Select The Module Name: ")

READ ("Type String", stA chModuleName)

CALL vUpper(stA chModuleName)

CLEAR BUFFERING

DISPLAY ("Sample Format: 29-12-2000")

DISPLAY ("Please Enter The Lecture Date: ")

READ ("Type String", stA chDate)

CALL vUpper(stA chDate)

CLEAR BUFFERING

DISPLAY ("")

DISPLAY ("Sample Format: 14:45")

DISPLAY ("Please Enter The Lecture Start Time: ")

READ ("Type String", stA chStartTime)

CALL vUpper(stA chStartTime)

CLEAR BUFFERING

DISPLAY ("")

DISPLAY ("Sample Format: 16:30")

DISPLAY ("Please Enter The Lecture End Time: ")

READ ("Type String", stA chEndTime)

CALL vUpper(stA chEndTime)

CLEAR BUFFERING

CLEAR SCREEN

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

IF (st intType==1) THEN

STRUCT stIntakeRecord st4

SET FILE \*fp4

fp4=OPEN FILE(dbIntakeRecord, "READ MODE")

DOWHILE(1)

READ FILE (st4, GET SIZE(st4), 1, fp4)

IF(READING(fp4)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(st chTP, st4 chUserID)

IF(intCompare==0) THAN

intCompare=COMPARE\_STRING(stA chIntakeCode, st4 chIntakeCode)

IF(intCompare==0) THEN

STRUCT stModuleRecord st5

SET FILE \*fp5

fp5=OPEN FILE(dbModuleRecord, "READ MODE")

DOWHILE(1)

READ FILE (st5, GET SIZE(st5), 1, fp5)

IF(READING(fp5)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(st chTP, st5 chUserID)

IF(intCompare==0) THEN

intCompare=COMPARE\_STRING(stA chModuleName, st5 chModuleName)

IF(intCompare==0) THEN

GET AND SET (stA chStudentName, st chName, 30)

GET AND SET (stA chStudentTP, st chTP, 16)

WRITE FILE (stA, GET SIZE(stA), 1, fp3)

END IF

END IF

END WHILEDO

FILE CLOSE(fp5)

END IF

END IF

FILE CLOSE(fp4)

FILE CLOSE(fp)

FILE CLOSE(fp3)

END

### Modify Attendance

Function\_vModIFyAttendance()

SET INT intCompare, intFound=0

STRUCT stAttendance st

STRUCT stAttendance st1

SET FILE \*fp, \*fp1

fp=OPEN FILE(dbAttendance, "READ MODE")

GET AND SET (stA chLecturerID, chUsername, 16)

CLEAR SCREEN

DISPLAY ("MODIFY ATTENDANCE")

DISPLAY ("Sample Format: 29-12-2000")

DISPLAY ("Please Enter The Attendance Taken Date: ")

READ ("Type String", stA chDate)

CALL vUpper(stA chDate)

CLEAR BUFFERING

DISPLAY ("Sample Format: 14:45")

DISPLAY ("Please Enter The Class Started Time: ")

READ ("Type String", stA chStartTime)

CALL vUpper(stA chStartTime)

CLEAR BUFFERING

DISPLAY ("Sample Format: 16:30")

DISPLAY ("Please Enter The Class Ended Time: ")

READ ("Type String", stA chEndTime)

CALL vUpper(stA chEndTime)

CLEAR BUFFERING

CLEAR SCREEN

DISPLAY ( MODIFY ATTENDANCE ")

DISPLAY ("TP NUMBER STATUS NAME:")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

intCompare=(COMPARE\_STRING(stA chStartTime,st chStartTime))+(COMPARE\_STRING(stA chEndTime,st chEndTime))+(COMPARE\_STRING(stA chDate,st chDate))+(COMPARE\_STRING(stA chLecturerID,st chLecturerID))

IF(intCompare==0) THEN

DISPLAY ("Type String", st chStudentTP)

IF(st intStatus==4) THEN

DISPLAY ("ABSENTR")

END IF

ELSE IF(st intStatus==3) THEN

DISPLAY ("ABSENT ")

END IF

ELSE IF(st intStatus==2) THEN

DISPLAY ("LATE ")

END IF

ELSE

DISPLAY ("PRESENT")

END IF

DISPLAY ("Type String",st chStudentName)

END IF

END IF

FILE CLOSE(fp)

DISPLAY ("")

fp=OPEN FILE(dbAttendance, "READ MODE")

fp1=OPEN FILE("dbTemp txt", "wb")

DISPLAY ("1 MARK ATTENDANCE0 FINISH")

DISPLAY ("Choose Your Option: ")

SET STRING chChoice

READ (Type\_Chacacter, chChoice)

CLEAR BUFFERING

SWITCH(chChoice) THEN

CASE '1':

BREAK

CASE '0':

CALL vLecturerMenu()

BREAK

END IF

DISPLAY ("Enter The Student's TP Number To ModIFy Attendance: ")

READ ("Type String", stA chStudentTP)

CALL vUpper(stA chStudentTP)

CLEAR BUFFERING

DOWHILE(1)

READ FILE (st1, GET SIZE(st1), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

SET INT intCompare

intCompare=(COMPARE\_STRING(stA chStudentTP, st1 chStudentTP))+(COMPARE\_STRING(stA chStartTime,st1 chStartTime))+(COMPARE\_STRING(stA chEndTime,st1 chEndTime))+(COMPARE\_STRING(stA chDate,st1 chDate))

IF(intCompare==0) THEN

intFound=1

DISPLAY ("")

DISPLAY ("1 PRESENT 2 LATE 3 ABSENT")

DISPLAY ("Choose Your Option: ")

SET STRING chChoice1

READ (Type\_Chacacter, chChoice1)

CLEAR BUFFERING

SWITCH(chChoice1) THEN

CASE '1':

st1 intStatus=1

BREAK

CASE '2':

st1 intStatus=2

BREAK

CASE '3':

st1 intStatus=3

BREAK

END IF

WRITE FILE (st1, GET SIZE(st1), 1, fp1)

END IF

ELSE

WRITE FILE (st1, GET SIZE(st1), 1, fp1)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

IF(intFound==0) THEN

DISPLAY ("Sorry, No Record Found!")

END IF

ELSE

fp=OPEN FILE(dbAttendance, "wb")

fp1=OPEN FILE("dbTemp txt", "READ MODE")

DOWHILE(1)

READ FILE (st1, GET SIZE(st1), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

WRITE FILE (st1, GET SIZE(st1), 1, fp)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

END

### View My Attendance

Function\_vViewAttendance()

SET STRING chModuleName LENGTH [16]

SET INT intCompare, intCompare\_2, intFound1=0

STRUCT stAttendance st

STRUCT stModuleRecord st1

SET FILE \*fp, \*fp1

fp=OPEN FILE(dbAttendance, "READ MODE")

fp1=OPEN FILE(dbModuleRecord, "READ MODE")

CLEAR SCREEN

DISPLAY ( VIEW MY ATTENDANCE ")

DISPLAY ("List Of Type String's Registered Modules:", chUsername)

DOWHILE(1)

READ FILE (st1, GET SIZE(st1), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(chUsername, st1 chUserID)

IF(intCompare==0) THEN

DISPLAY ("Type String", st1 chModuleName)

intFound1=1

END IF

END IF

IF(intFound1==0) THEN

DISPLAY ("NOT FOUND!")

END IF

FILE CLOSE(fp1)

DISPLAY ("Enter The Module's Name To View Attendance: ")

READ ("Type String", chModuleName)

CALL vUpper(chModuleName)

CLEAR BUFFERING

CLEAR SCREEN

DISPLAY ( VIEW MY ATTENDANCE ")

DISPLAY ("MODULE: Type String", chModuleName)

DISPLAY ("TP NUMBER: Type String", chUsername)

DISPLAY ("DATESTARTENDSTATUS")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(chUsername,st chStudentTP)

intCompare\_2=COMPARE\_STRING(chModuleName,st chModuleName)

IF((intCompare==0)(intCompare\_2==0)) THEN

DISPLAY ("")

DISPLAY ("Type String", st chDate)

DISPLAY ("Type String", st chStartTime)

DISPLAY ("Type String", st chEndTime)

IF(st intStatus==4) THEN

DISPLAY ("ABSENTR")

END IF

ELSE IF(st intStatus==3) THEN

DISPLAY ("ABSENT ")

END IF

ELSE IF(st intStatus==2) THEN

DISPLAY ("LATE ")

END IF

ELSE

DISPLAY ("PRESENT")

END IF

END IF

END IF

FILE CLOSE(fp)

DISPLAY ("")

END

## Users

### Add Lecturer & Student

Function\_vAddLecturer()

CLEAR SCREEN

SET FILE \*fp

STRUCT stLogin st

st intType = 2

fp=OPEN FILE(dbLoginList, "READ AND WRITE MODE")

DISPLAY ( ADD NEW LECTURER")

DISPLAY ("Enter The Lecturer TP Number: ")

READ ("Type String", st chTP)

CALL vUpper(st chTP)

CLEAR BUFFERING

DISPLAY ("Enter The Lecturer Password: ")

READ ("Type String", st chPass)

CALL vUpper(st chPass)

CLEAR BUFFERING

DISPLAY ("Enter The Lecturer Name: ")

READ ("%[^]%\*c", st chName)

CALL vUpper(st chName)

CLEAR BUFFERING

WRITE FILE (st, GET SIZE(st), 1, fp)

FILE CLOSE(fp)

END

### Search Lecturer & Student

Function\_vSearchLecturer()

CLEAR SCREEN

SET FILE \*fp, \*fp1, \*fp2

STRUCT stLogin st

STRUCT stIntakeRecord st1

STRUCT stModuleRecord st2

SET INT intFound=0, intFound1=0, intFound2=0, intCompare, intCompare1, intCompare2

SET STRING chTP LENGTH [16]

fp=OPEN FILE(dbLoginList, "READ MODE")

fp1=OPEN FILE(dbIntakeRecord, "READ MODE")

fp2=OPEN FILE(dbModuleRecord, "READ MODE")

DISPLAY ("SEARCH LECTURER")

DISPLAY ("Enter The Lecturer TP Number: ")

READ ("Type String", chTP)

CALL vUpper(chTP)

CLEAR BUFFERING

CLEAR SCREEN

DISPLAY ("SEARCH LECTURER")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

intCompare=COMPARE\_STRING(chTP, st chTP)

IF(intCompare==0) THEN

DISPLAY ("TP NUMBER:")

DISPLAY ("Type String", st chTP)

DISPLAY ("NAME:")

DISPLAY ("Type String", st chName)

DISPLAY ("PASSWORD:")

DISPLAY ("Type String", st chPass)

intFound=1

END IF

END IF

IF(intFound==0) THEN

DISPLAY ("Sorry, No Record Found!")

DISPLAY ("")

FILE CLOSE(fp)

return

END IF

FILE CLOSE(fp)

DISPLAY ("INTAKE CODE:")

DOWHILE(1)

READ FILE (st1, GET SIZE(st1), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

intCompare1=COMPARE\_STRING(chTP, st1 chUserID)

IF(intCompare1==0) THEN

DISPLAY ("Type String", st1 chIntakeCode)

intFound1=1

END IF

END IF

IF(intFound1==0) THEN

DISPLAY ("NOT FOUND!")

END IF

FILE CLOSE(fp1)

DISPLAY ("MODULE NAME:")

DOWHILE(1)

READ FILE (st2, GET SIZE(st2), 1, fp2)

IF(READING(fp2)) THEN

BREAK

END IF

intCompare2=COMPARE\_STRING(chTP, st2 chUserID)

IF(intCompare2==0) THEN

DISPLAY ("Type String", st2 chModuleName)

intFound2=1

END IF

END IF

IF(intFound2==0) THEN

DISPLAY ("NOT FOUND!")

END IF

FILE CLOSE(fp2)

END

### Modify Lecturer & Student

Function\_vModIFyLecturer()

CLEAR SCREEN

SET FILE \*fp, \*fp1

STRUCT stLogin st

SET INT intFound=0

SET STRING chTP LENGTH [16]

st intType=2

fp=OPEN FILE(dbLoginList, "READ MODE")

fp1=OPEN FILE("dbTemp txt", "wb")

DISPLAY ("MODIFY LECTURER")

DISPLAY ("Enter The Lecturer's TP Number You Want To ModIFy: ")

READ ("Type String", chTP)

CALL vUpper(chTP)

CLEAR BUFFERING

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

SET INT intCompare

intCompare=COMPARE\_STRING(chTP, st chTP)

IF(intCompare==0) THEN

intFound=1

DISPLAY ("Enter The New Lecturer TP Number: ")

READ ("Type String", st chTP)

CALL vUpper(st chTP)

CLEAR BUFFERING

DISPLAY ("Enter The New Lecturer Password: ")

READ ("Type String", st chPass)

CALL vUpper(st chPass)

CLEAR BUFFERING

DISPLAY ("Enter The New Lecturer Name: ")

READ ("%[^]%\*c", st chName)

CALL vUpper(st chName)

CLEAR BUFFERING

WRITE FILE (st, GET SIZE(st), 1, fp1)

END IF

ELSE

WRITE FILE (st, GET SIZE(st), 1, fp1)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

IF(intFound==0) THEN

DISPLAY ("Sorry, No Record Found!")

END IF

ELSE

fp=OPEN FILE(dbLoginList,"wb")

fp1=OPEN FILE("dbTemp txt","READ MODE")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

WRITE FILE (st, GET SIZE(st), 1, fp)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

END

### Delete Lecturer & Student

Function\_vDeleteLecturer()

CLEAR SCREEN

SET FILE \*fp,\*fp1

STRUCT stLogin st

SET INT intFound=0

SET STRING chTP LENGTH [16]

st intType=2

fp=OPEN FILE(dbLoginList, "READ MODE")

fp1=OPEN FILE("dbTemp txt", "wb")

DISPLAY ("DELETE LECTURER")

DISPLAY ("Enter The Lecturer's TP Number You Want To Delete: ")

READ ("Type String", chTP)

CALL vUpper(chTP)

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

SET INT intCompare

intCompare=COMPARE\_STRING(chTP, st chTP)

IF(intCompare==0) THEN

intFound=1

END IF

ELSE

WRITE FILE (st, GET SIZE(st), 1, fp1)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

IF(intFound==0) THEN

DISPLAY ("Sorry, No Record Found!")

END IF

ELSE

fp=OPEN FILE(dbLoginList, "wb")

fp1=OPEN FILE("dbTemp txt", "READ MODE")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp1)

IF(READING(fp1)) THEN

BREAK

END IF

WRITE FILE (st, GET SIZE(st), 1, fp)

END IF

END IF

FILE CLOSE(fp)

FILE CLOSE(fp1)

END

### View All Lecturer & Students

Function\_vViewAllLecturer()

SET FILE \*fp

STRUCT stLogin st

fp=OPEN FILE(dbLoginList, "READ MODE")

CLEAR SCREEN

DISPLAY ( VIEW ALL LECTURER ")

DISPLAY ("TP NUMBER: NAME:")

DOWHILE(1)

READ FILE (st, GET SIZE(st), 1, fp)

IF(READING(fp)) THEN

BREAK

END IF

IF (st intType==2) THEN

DISPLAY ("Type String",st chTP)

DISPLAY ("Type String",st chName)

END IF

END IF

FILE CLOSE(fp)

END

# conclusion

# references