DEBUG07-01

// Each time a salesperson sells a car at the

// Pardeeville New and Used Auto Dealership,

// a record is created containing the salesperson's

// name and the amount of the sale.

// Sales of new and used cars are kept in separate files,

// sorted by salesperson ID number.

// Management has requested a merged file so that

// all of a salesperson's sales (both new and used cars)

// are displayed together. The following code is intended

// to merge the files.

start

Declarations

string newSalesperson

num newAmount

string usedSalesperson

num usedAmount

string bothAtEof = "N"

string HIGH\_NAME = "ZZZZZ"

InputFile newSales

InputFile usedSales

OutputFile allSales // should use camel casing

getReady()

while bothAtEof = "N" // starting value should be N

detailLoop()

endwhile

finish()

stop

getReady()

open newSales "NewSales.dat"

open usedSales "UsedSales.dat"

open allSales "AllSales.dat"

input newSalesperson, newAmount from newSales

if eof then

newSalesperson = HIGH\_NAME // should be newSalesperson

endif

input usedSalesperson, usedAmount from usedSales

if eof then

usedSalesperson = HIGH\_NAME

endif

if newSalesperson = HIGH\_NAME AND usedSaleserson = HIGH\_NAME then

bothAtEof = "Y"

endif

return

detailLoop()

if newSalesperson > usedSalesperson then

output usedSalesperson, usedAmount to allSales

input usedSalesperson, usedAmount from usedSales // need to read used after outputing used

if eof then

usedSalesperson = HIGH\_NAME

endif

else

output newSalesperson, newAmount to allSales

input newSalesperson, newAmount from newSales // need to read new after outputing new

if eof then

newSalesperson = HIGH\_NAME

endif

endif

if newSalesperson = HIGH\_NAME AND usedSalesperson = HIGH\_NAME then

bothAtEof = "Y"

endif

return

finish()

close newSales

close usedSales

close allSales

return

DEBUG07-02

// The two senior class homerooms at Littleville High School

// are having a fundraising drive for the prom. Each time a student

// solicits a contribution, a record is created with the

// student's name and the value. Two files have been created for

// Homeroom A and Homeroom B. Each file is sorted in contribution

// value order from highest to lowest. This program merges the two files.

start

Declarations

string roomAName

num roomAValue

string roomBName

num roomBValue

string bothFilesDone = "N"

num HIGH\_VALUE = 999999

InputFile roomAFile

InputFile roomBFile

OutputFile mergedFile // should be declared as an OutputFile

getReady()

while bothFilesDone = "N"

detailLoop()

endwhile

allDone()

stop

getReady()

open roomAFile "roomAFile.dat"

open roomBFile "roomBFile.dat"

open mergedFile "mergedFile.dat"

readA()

readB()

checkBoth()

return

readA()

input roomAName, roomAValue from roomAFile

if eof then

roomAValue = HIGH\_VALUE

endif

return

readB()

input roomBName, roomBValue from roomBFile

if eof then

roomBValue = HIGH\_VALUE

endif

return

checkBoth()

if roomAValue = HIGH\_VALUE AND roomBValue = HIGH\_VALUE then

bothFilesDone = "Y"

endif

return

detailLoop()

if roomAValue > roomBValue then

output roomBName, roomBValue to mergedFile

readB() // must read from file that was output from

else

output roomAName, roomAValue to mergedFile

readA() // must read from file that was output from

endif

checkBoth() // need to call checkBoth or you'll have an infinite loop

return

allDone()

close roomAFile

close roomBFile

close mergeFile

return

DEBUG07-03

// Cooper College maintains a master file of students and credits

// earned. Each semester the master is updated with a transaction

// file that contains credits earned during the semester.

// Each file is sorted in Student ID number order.

start

Declarations

num masterID

string masterName

num masterCredits

num transID

num transCredits

string bothDone = "N"

num HIGH\_VALUE = 999999

InputFile master

InputFile trans

OutputFile newMaster

getReady()

while bothDone = "N"

detailLoop()

endwhile

allDone()

stop

getReady()

open master "studentFile.dat"

open trans "semesterCredits.dat"

open newMaster "updatedStudentFile.dat"

readMaster()

readTrans()

checkBoth()

return

readMaster()

input masterID, masterName, masterCredits from master

if eof then

masterID = HIGH\_VALUE

endif

return

readTrans()

input transID, transCredits from trans

if eof then

transID = HIGH\_VALUE

endif

return

checkBoth()

if masterID = HIGH\_VALUE AND transID = HIGH\_VALUE then

bothDone = "Y"

endif

return

detailLoop()

if masterID = transID then

match()

else

if masterID > transID then

noMasterForTrans()

else

noTransForMaster()

endif

endif

checkBoth() // call checkBoth or cause an infinite loop

return

match()

masterCredits = masterCredits + transCredits // credits must be added

output masterID, masterName, masterCredits to newMaster

readMaster()

readTrans()

return

noMasterForTrans()

output "No master file record matches transaction ", masterID

readMaster() // must call readMaster again

readTrans()

return

noTransForMaster()

output masterID, masterName, masterCredits to newMaster

return

allDone()

close master

close trans

close newMaster

return