```
!-------------------------------------------------------------
!>  Lab 1 - Linked List Management Program (Fortran)
!   Blake Lucas and Luke Reddick
!   COSC 30403-035
!   7 February 2017
!
!>  This program reads in a input text file and performs
!   operations--insert, delete, update, and print (in various
!   forms)--on a singly linked list sorted ascendingly by the
!   employee's ID number.
!-------------------------------------------------------------

program Main
    use EmployeeModule
    implicit none

    ! Variable declarations
    INTEGER  :: openStat, lineStat , id, outStat    ! *Stat used for IO control, id for each command
    REAL     ::  pay                                ! Update salary variable
    CHARACTER(LEN=23) :: en, dn, ln, jt             ! Variable parameter strings
    CHARACTER(LEN=2) :: cmd                         ! String for each command
    CHARACTER(63) :: line                           ! String for each new line in the file
    type(Employee), pointer :: listHead             ! Pointer for the linked list header
    type(Employee), pointer :: n                    ! Pointer for a new employee
    character(32), dimension(5) :: parts            ! Array for the split line on insert

    ! Allocate the list header
    allocate(listHead)
        listHead%name = "HEADER"
        listHead%id = -1                            ! -1 so it will always be first in the ordered linked list.
        listHead%deptName = "HEADER"
        listHead%position = "HEADER"
        listHead%salary = -1.00
        nullify(listHead%next)

    ! Open Lab1Ans file for the output
    open(unit=1, file="./Lab1Ans.txt", iostat=outStat)
    ! Check to ensure the open was successful.
    if (outStat .ne. 0) write(*,*) "Error opening output file."
    ! Open the Lab1Data input file
    OPEN( unit=2 , file="./Lab1Data.txt", iostat=openStat)
    ! Check to ensure the open was successful.
    if (openStat /= 0) WRITE(*,*) "File cannot be opened"
    do
        ! Read each line from file
        READ(2,'(A63)', iostat=lineStat) line
        if (lineStat < 0) then
            CLOSE(2)
            call printAllEmps(listHead)             ! Print all employess on EOF.
            exit
        end if

        READ(line, '(A2, 1X)') cmd                  ! Read the line's command and switch
        select case (cmd)
            case ("IN")
```

```
56                read(line, '(A63)')
57                parts = splitParts(line)            ! Get all parts of each line
58                allocate(n)                         ! Allocate a new node for the new Employee
59                    n%name = parts(2)
60                    read(parts(1), '(i32)') n%id
61                    n%deptName = parts(3)
62                    n%position = parts(4)
63                    read(parts(5), '(F31.2)') n%salary
64                    call insert(listHead, n)        ! Insert the new Employee
65            case ("DE")
66                read(line, '(3X, I8)') id           ! Read the ID
67                write(1,*), "Deleting employee ", id
68                call delete(listHead, id)           ! Delete Employee with the corresponding
  ID
69            case ("UN")
70                read(line, '(3X, I8, 29X, A12)') id, ln     ! Read the ID and new last name
71                write(1,*), "Updating employee last name . . ."
72                call updateLN(listHead, id, ln)     ! Update the specified Employee's last
  name
73            case ("UT")
74                read(line, '(3X, I8, 29X, A12)') id, jt ! Read the ID and new title
75                write(1,*), "Updating employee job title . . ."
76                call updateTitle(listHead, id, jt)  ! Update the specified Employee's title
77            case ("UR")
78                read(line, '(3X, I8, 31X, F7.2)') id, pay   ! Read the ID and the new salary
79                write(1,*), "Updating employee salary . . ."
80                call updatePay(listHead, id, pay)   ! Update the specified Employee's salary
81            case ("UD")
82                read(line, '(3X, I8, 29X, A23)') id, dn     ! Read the ID and new department
83                write(1,*), "Updating employee department name . . ."
84                call updateDept(listHead, id, dn)   ! Update the specified Employee's
  department
85            case ("PA")
86                call printAllEmps(listHead)         ! Print all Employees
87            case ("PI")
88                read(line, '(3X, I8)') id           ! Read the ID
89                write(1,*), "Printing information for ", id
90                call printData(findById(listHead, id))      ! Print the data for the
  corresponding Employee, if available
91            case ("PD")
92                read(line, '(3X, A12)') dn          ! Read the department name
93                write(1,*), "Displaying all employees in ", dn
94                call printDept(listHead, dn)        ! Print all Employees in the department
95                write(1,*), "All employees in ",dn, " displayed"
96        END SELECT
97    END DO
98    close(1)                                        ! Close the output file
99
100   contains
101   function splitParts(line) result(parts)         ! Substrings each line into a length 5
  array
102       character(63) :: line
103       character(31), dimension(5) :: parts
104       parts(1) = line(4:11)                       ! ID
105       parts(2) = line(13:24)                      ! Name
106       parts(3) = line(25:40)                      ! Department
107       parts(4) = line(41:58)                      ! Position
```

```
108         parts(5) = line(58:63)                      ! Salary
109     end function splitParts
110 end program Main
111
```