```c
/*
 --------------------------------------------------------------------
 PROGRAM NAME: Lab 1, Read Arithmetic Expressions
 PROGRAMMER:   Samuel Jentsch
 CLASS:        CSC 214, Spring 2014
 INSTRUCTOR:   Dr. Strader
 DATE STARTED: January 20, 2014
 DUE DATE:     January 22, 2014
 REFERENCES:   Computer Organization and Architecture by Null and Lobor
 Beginning C by Ivor Horton
 Dr. Strader: assignment information sheet

 PROGRAM PURPOSE:
 a. This program reads an arithmetic expression from a data file
    character by character. It converts the character's into
    float values, performs the operations specfied by the file,
    and prints the value of the expression.

 VARIABLE DICTIONARY:
 OPERATORS - char[] contains the possible operators that will be
               in the file.
 NUM_OPERATORS - the number of operator types present in the file.

 FILES USED:
 lab1.dat - a file containing arithmetic expressions separated by
           semicolons.
 --------------------------------------------------------------------
 */

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>

#define NUM_OPERATORS 4

//Functions
int isOperator(char c);
float convertChars(char number[], int numberSize, int wholeDigits, int decimalDigits, int sign);
```

```c

const char OPERATORS[] = {'+', '-', '.', ';'};

int main(int argc, const char * argv[]) {

    FILE *dataFile;
    dataFile = fopen("../instr/lab1.dat", "r");
    if(dataFile == NULL) {
        printf("Error, file not found.");
        exit(1);
    }

    //1 for positive, -1 for negative.
    int signInt = 1;

    //The number of digits encountered
    int wholeDigits = 0;
    int fractionDigits = 0;

    //The current index within the number (char array)
    int numberIndex = 0;

    //Boolean (0 or 1) for decimal point encounter
    int decimalEncountered = 1;

    //Index of operator read from file
    //returned by isOperator()
    int opIndex = -1;

    //If 1, move on to new number.
    int stateChange = 0;

    //Character read from the file.
    char inputChar = 0;

    //Holds the number as chars in a char array.
    char number[256];

    float sum = 0;
```

```
//Read a character from the file and display it
while((inputChar = fgetc(dataFile)) != EOF) {

    if(inputChar != ' ' && inputChar != '\n') {
        //Skip spaces and new lines.
        if(isdigit(inputChar) != 0) {
            //Character is a digit.
            number[numberIndex] = inputChar;
            numberIndex++;
            if (decimalEncountered == 1) {
                wholeDigits++;
            } else {
                fractionDigits++;
            }
        }//end if digit
        else if((opIndex = isOperator(inputChar)) != -1) {
            if(opIndex == 0 || opIndex ==  1) {
                if (stateChange == 0) {
                    stateChange = 1;
                } else {
                    sum += convertChars(number, numberIndex, wholeDigits, fractionDigits, signInt);
                    decimalEncountered = 1;
                    signInt = 1;
                    wholeDigits = 0;
                    fractionDigits = 0;
                    numberIndex = 0;
                }
            }//end if + or -
            switch(opIndex) {
                case 0:
                    signInt = 1;
                    break;
                case 1:
                    signInt = -1;
                    break;
                case 2:
                    decimalEncountered = 0;
```

```
                        number[numberIndex] = '.';
                        numberIndex++;
                        break;
                    case 3:
                        sum += convertChars(number, numberIndex, wholeDigits, fractionDigits, signInt);
                        printf("The value is %.4f.\n", sum);
                        decimalEncountered = 1;
                        signInt = 1;
                        wholeDigits = 0;
                        fractionDigits = 0;
                        numberIndex = 0;
                        sum = 0;
                        break;
                    default:
                        printf("Error");
                }//end switch if operator

            }//end if operator

        }//end if not space

    }//end while

    fclose(dataFile);

    dataFile = NULL;

    return 0;
}


int isOperator(char c) {
    /*-----------------------------------*/
    //Return index of operator if
    // char c is an operator, returns -1
    //if not.
    //Preconditions: char c passed as
    //parameter.
    //Postconditions: Index of operator
    //returned if c is an operator, -1
    //returned if not.
    /*------------------------------------*/
    int returnValue = -1;
    int i = 0;

    for(i = 0; (i < NUM_OPERATORS && returnValue == -1); i++) {
        if(c == OPERATORS[i]) {
            returnValue = i;
        }
    }//end for

    return returnValue;
}//isOperator

float convertChars(char number[],int numberSize, int wholeDigits, int decimalDigits, int sign) {
    /*-------------------------------------------------*/
    //Return the float value of a char array (string)
    //representation of a number. This method uses
    //the parameters passed to convert the char[] to
    //a float value and returns the value.
    //Preconditions: number passed as char array,
    //numberSize representing the total number of
    //elements in number[], wholeDigits containing the
    //number of digits before a decimal point in the
    //number, and decimal digits containing the number
    //of digits after the decimal point in the number.
    //Sign containing either 1 or -1 depending on
    //the sign of the number.
    //Postconditions: The number char array is converted
    //to a float value and returned.
    /*-------------------------------------------------*/
    float floatValue = 0;
    float powerOfTen = 1;
    int digit = 0;
    int decimalBool = 1;
    int totalDecimal = decimalDigits;

    int i = 0;
    int j = 0;

    for(i = 0; i < numberSize; i++) {

        if (number[i] != '.' && decimalBool == 1) {
            digit = number[i] - '0';
            powerOfTen = 1;
            for(j = 0; j < wholeDigits - 1; j++) {
                powerOfTen *= 10;
            }
            wholeDigits--;
            floatValue += (digit*powerOfTen);
        } else if(decimalBool == 0) {
            digit = number[i] - '0';
            powerOfTen = 1;
            for(j = decimalDigits - 1; j < totalDecimal; j++) {
                powerOfTen /= 10;
            }
            decimalDigits--;
            floatValue += (digit*powerOfTen);
        }
        else {
            decimalBool = 0;
        }

    }//end for

    floatValue *= sign;
    return floatValue;
}
```