```c
/*--------------------------------
 *  Christopher Fields
 *  90.267 C Programming
 *  Programming Assignment #2
 *  3/11/2014
 *  Arrays; Saving & Analysis
 *--------------------------------*/
/*
 * This program begins in a data collection loop.  Prompting for two related
 * datapoints (ID, Score), saving each to it's respective array.  Based upon
 * the test data, I will assume both arrays are integers, studentID expects a 4
 * digit integer, and studentScore expects an integer 0 to 100.  The data
 * collection loop will end when either the ID # 0 is encountered, or when the
 * array becomes full (50 items).
 *
 * From inputed data, create array studentGrade, converting scores to grades by
 * iterating over studentScore and using if / else loops to define letter grades.
 * Meanwhile generate sum of all grades, and increment individual variables for
 * each grade.
 *
 * Finally the program will generate & print a report stating every ID, Score,
     Grade,
 * the average numeric score, the number of students who scored above average,
 * number of scores in each alphabetical category [A, B, C, D, F], and finally
 * the ID, Score, and Lettergrade of the highest
 * score.
 */

#include <stdio.h>
#define MAXIMUMSTUDENTS 50  // Size of Array

int main(void) {

    int studentID[MAXIMUMSTUDENTS], studentScore[MAXIMUMSTUDENTS];
    int i = 0;
    int studentCount = 0;
    int sum = 0;
    float average = 0.0;
    int highScore = 0, highScoreIndex = 0;
    int aboveAverage = 0;
    int a = 0, b = 0, c = 0, d = 0, f = 0;


    //  Welcome statements
    printf ("Welcome, Please enter upto %i pairs of Student ID's & scores.\n",
        MAXIMUMSTUDENTS);
    printf ("To halt the data entry process, enter 0 for the Student ID.\n\n");


    //  Data collection loop
    do {
        printf ( "\tStudent's ID: " );
        scanf  ( "%i", &studentID[i] );

        //  Exit loop when student ID equals 0
        if ( studentID[i] == 0 ) {
```

```
            printf("End of data collection.\n\n");
            break;
        }

        printf ( "\tEnter Grade: " );
        scanf  ( "%i", &studentScore[i] );

        ++i;
    }   //  Quit when iterations exceeds array size
    while ( i < MAXIMUMSTUDENTS );

    studentCount = i;                   //  # of students used

    if ( studentCount > 0 ) {           //  Skip program if first ID = 0.

        for (i = 0; i < studentCount; ++i)
            if (studentScore[i] > highScore) {
                highScore = studentScore[i];
                highScoreIndex = i;     //  Return: index of highest score
            }

        char studentGrade[MAXIMUMSTUDENTS]; //  Array for letter grades

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

            sum += studentScore[i];         // build sum of all scores

            if ( studentScore[i] >= 90 && studentScore[i] <= 100 ) {
                studentGrade[i] = 'A';
                a += 1;
            }
            else if ( studentScore[i] >= 80 && studentScore[i] < 90 ) {
                studentGrade[i] = 'B';
                b += 1;
            }
            else if ( studentScore[i] >= 70 && studentScore[i] < 80 ) {
                studentGrade[i] = 'C';
                c += 1;
            }
            else if ( studentScore[i] >= 60 && studentScore[i] < 70 ) {
                studentGrade[i] = 'D';
                d += 1;
            }
            else {                      //  Provided data 0 < score <= 100
                studentGrade[i] = 'F';
                f += 1;
            }

        }

        average = (float)sum/ studentCount;

        for (i = 0; i < studentCount; ++i)
            if (studentScore[i] > average) {
                aboveAverage += 1;
            }
```

```
        //  Print out (3) arrays
        printf ("Data Report:\n");
        printf ("\t--------------------\n");
        printf ("\t ID : Score : Grade\n");
        printf ("\t--------------------\n");
        for (i = 0; i < studentCount; ++i){
            printf ("\t%4i   %3i      %1c \n", studentID[i], studentScore[i],
                studentGrade[i]);
        }
        printf ("\t--------------------\n\n");

        //  Print Out the Results
        printf ("The average test score in the set was: %2.2f\n", average);

        printf ("The number of over average scores was: %i\n\n", aboveAverage);

        printf ("The grade distribution is:\n");

        printf ("\tA: %i\n\tB: %i\n\tC: %i\n\tD: %i\n\tF: %i\n", a, b, c, d, f);

        printf ("Student # %i received the highest score of: %i, for a grade of:
            %c\n", studentID[highScoreIndex], studentScore[highScoreIndex],
            studentGrade[highScoreIndex]);
    }


    printf ("End of program.\n");
}
```