```
#include <fstream>
#include <iostream>
#include <iomanip>
using namespace std;

// This program reads records from a file. The file contains the
// following: student’s name, two test grades and final exam grade.
// It then prints this information to the screen.

// Michael Steele

const int NAMESIZE = 15;
const int MAXRECORDS = 50;

struct Grades // declares a structure
{
        char name[NAMESIZE + 1];
        int test1;
        int test2;
        int final;
        char letter;
};

typedef Grades gradeType;
// This makes gradeType a data type
// that holds MAXRECORDS
// Grades structures.

// FIll IN THE CODE FOR THE PROTOTYPE OF THE FUNCTION ReadIt
// WHERE THE FIRST ARGUMENT IS AN INPUT FILE, THE SECOND IS THE
// ARRAY OF RECORDS, AND THE THIRD WILL HOLD THE NUMBER OF RECORDS
// CURRENTLY IN THE ARRAY.
void readIt(ifstream &,gradeType *, int &);
char letter(int test1,int test2, int final);


int main()
{
        ifstream indata;

        indata.open("graderoll.txt");

        int numRecord;        // number of records read in
        gradeType studentRecord[MAXRECORDS];

        if (!indata)
        {
                cout << "Error opening file. \n";
                cout << "It may not exist where indicated" << endl;

                return 1;
        }

        readIt(indata,studentRecord,numRecord);

        // output the information
        for (int count = 0; count < numRecord; count++)
        {
                cout << studentRecord[count].name << setw(10)
                         << studentRecord[count].test1
                         << setw(10) << studentRecord[count].test2;
                cout << setw(10) << studentRecord[count].final ;
                cout << "          " << "Final grade " << studentRecord[count].letter << endl;
        }

        return 0;
}

//*************************************************************
//      readIt
//
//      task:     This procedure reads records into an array of
//                records from an input file and keeps track of the
//                total number of records
//      data in:  data file containing information to be placed in
//                the array
//      data out: an array of records and the number of records
//
//*************************************************************

void readIt(ifstream &indata, gradeType *gradeRec ,int &total)// FILL IN THE CODE FOR THE
FORMAL PARAMETERS AND THEIR
                // DATA   TYPES.
                // inData, gradeRec and total are the formal parameters
                // total is passed by reference)
```

```
{
        total = 0;

        indata.get(gradeRec[total].name, NAMESIZE);

        while (indata)
        {
                // FILL IN THE CODE TO READ test1
                indata >> gradeRec[total].test1;

                // FILL IN THE CODE TO READ test2
                indata >> gradeRec[total].test2;

                // FILL IN THE CODE TO READ final
                indata >> gradeRec[total].final;
      gradeRec[total].letter =
letter(gradeRec[total].test1,gradeRec[total].test2,gradeRec[total].final);
                total++;        // add one to total

                // FILL IN THE CODE TO CONSUME THE END OF LINE
      char eat;
                indata.get(eat);

                // FILL IN THE CODE TO READ name
                indata.get(gradeRec[total].name, NAMESIZE);


        }

}
char letter(int test1,int test2, int final)
{
        int avg = 0.3 * test1 + 0.3 * test2 +0.4 * final;
        if(avg >= 90)
                return 'A';
        else if(avg >= 80)
                return 'B';
        else if(avg >=  70)
                return 'C';
        else if(avg >= 60)
                return 'D';
   else
      return 'F';
}
```