```c
/* J Hundley - grades with strings
File name: grades_2D.c
Program: Students' names and HW grades
------------------------------------------------------------
Read the names and grades into 2D arrays and print report.
------------------------------------------------------------
*/
#include <stdio.h>
#include <string.h>
#define  MAX_STUDENT  20 // max number of students in class
#define  NUM_HWS      10 // number of assignments
#define  NAME_LENGTH  11 // number of characters in a name+1
#define  TITLE_LENGTH 41 // number of characters in the title+1

//-------------------
// function prototypes
//-------------------
   int  readData    ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] ); //read data into arrays
   void printReport ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
   void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
   void printHeader ( char title[] );                                    //print headers

//-------------------
// main
//-------------------
    int main()
   {
      // declare and initialize variables
      char title[TITLE_LENGTH];             // file/report title
      char fname[MAX_STUDENT][NAME_LENGTH], // first name
           lname[MAX_STUDENT][NAME_LENGTH]; // last name
      int  grades[MAX_STUDENT][NUM_HWS];        // homework grades
      int  nStu;                            // number of students in class

         // INPUT - read names and grades into arrays--------------------
      nStu = readData( title, fname, lname, grades );
      if ( nStu < 1 ) printf( "No data available\n" );
      else
         // OUTPUT - print report ---------------------------------------
         printReport( title, fname, lname, grades, nStu );

      return 0;
   }
//----------------------
// read data into arrays
//----------------------
   int readData( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] )
   {
      int n = 0; // game counter
      int h;     // HW counter
      FILE *inp;
      inp = fopen( "HW_Grades.txt", "r" );
      if ( inp == NULL )
         printf( "Open file error\n" );
      else
      {
         fgets( title, TITLE_LENGTH, inp );
         while ( fscanf( inp, "%s %s", &fname[n], &lname[n] ) !=EOF )  // read one row (name) at a time
         {
```

```
INPUT FILE -- HW_Grades.txt

COMP1200 Fall 2009 Homework Grades
Susan williams 100 100 98 95 97 98 95 95 95 36
Judith Holliday 100 100 100 97 100 99 100 100 100 93
Brenda Young 96 89 54 98 77 49 97 47 98 79
Edward Moore 100 100 100 100 100 99 100 100 93 100
Thomas Edwards 100 100 98 98 97 99 99 99 95 85
Robert Smyth 100 100 100 94 100 84 99 65 98 55
Carolyn Sims 100 100 88 98 100 84 100 85 100 65
David Simpson 100 65 25 50 97 65 80 10 30 65
Jefferson Dykes 100 100 100 74 97 89 100 100 100 85
Michael Brock 100 80 88 78 80 93 72 79 65 40
Mitchell Cleveland 100 80 100 95 95 99 100 50 80 85
```

```c

            for ( h=0; h<NUM_HWS; h++ ) fscanf( inp, "%d", &grades[n][h] );

            n++;
         }
      }
      return n;
   }

//---------------------
// print the report
//---------------------
   void printReport( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
   {
      printHeader( title );
      printDetails( fname, lname, grades, nStu );
   }
//---------------------
// print body of report
//---------------------
   void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
   {
      int n,h;
      int HW_Totals[NUM_HWS] = {0};
      int stu_Total;

      for ( n=0; n<nStu; n++ )
      {
         stu_Total = 0;
         printf( "%-12s %-12s ", fname[n], lname[n] );
         for ( h=0; h<NUM_HWS; h++ )
         {
            printf( "%5d", grades[n][h] );
            stu_Total += grades[n][h]; // student's grades accumulator
            HW_Totals[h] += grades[n][h]; // HW accumulator
         }
         printf( " -- %4.1f\n", (double)stu_Total/NUM_HWS );
      }

      printf( "------------------------------------------------------------------------------------\n" );

         // print average by assignments
      printf( "Assignment Average         " );
      for ( h=0; h<NUM_HWS; h++ ) printf( "%5.1f", (double)HW_Totals[h]/nStu );
      printf( "\n" );
   }

//-------------------
// print headers
//-------------------
    void printHeader( char title[] )
   {
      printf( "                       %s", title );
      printf( "------------------------------------------------------------------------------------\n" );
      printf( "                              1    2    3    4    5    6    7    8    9   10      Ave\n" );
      printf( "------------------------------------------------------------------------------------\n" );
   }

```

```
OUTPUT --
                       COMP1200 Fall 2009 Homework Grade
------------------------------------------------------------------------------------
                              1    2    3    4    5    6    7    8    9   10      Ave
------------------------------------------------------------------------------------
Susan        williams       100  100   98   95   97   98   95   95   95   36 -- 90.9
JUdith       Holliday       100  100  100   97  100   99  100  100  100   93 -- 98.9
Brenda       Young           96   89   54   98   77   49   97   47   98   79 -- 78.4
Edward       Moore          100  100  100  100  100   99  100  100   93  100 -- 99.2
Thomas       Edwards        100  100   98   98   97   99   99   99   95   85 -- 97.0
Robert       Smyth          100  100  100   94  100   84   99   65   98   55 -- 89.5
Carolyn      Sims           100  100   88   98  100   84  100   85  100   65 -- 92.0
David        Simpson        100   65   25   50   97   65   80   10   30   65 -- 58.7
Jefferson    Dykes          100  100  100   74   97   89  100  100  100   85 -- 94.5
Michael      Brock          100   80   88   78   80   93   72   79   65   40 -- 77.5
Michell      Cleveland      100   80  100   95   95   99  100   50   80   85 -- 88.4
------------------------------------------------------------------------------------
 Assignment Average          99.6 92.2 86.5 88.8 94.5 87.1 94.7 75.5 86.7 71.6
```

grades_2D.c

```
/* J Hundley - grades with strings
File name: grades_2D_cat.c
Program: Students' names and HW grades
----------------------------------------------------------
Read the names and grades into 2D arrays and print report.
----------------------------------------------------------
*/
#include <stdio.h>
#include <string.h>
#define  MAX_STUDENT  20 // max number of students in class
#define  NUM_HWS      10 // number of assignments
#define  NAME_LENGTH  11 // number of characters in a name+1
#define  TITLE_LENGTH 41 // number of characters in the title+1

//-------------------
// function prototypes
//-------------------
   int  readData    ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] ); //read data into arrays
   void printReport ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
   void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
   void printHeader ( char title[] );                                               //print headers

//-------------------
// main
//-------------------
    int main()
   {
      // declare and initialize variables
      char title[TITLE_LENGTH];             // file/report title
      char fname[MAX_STUDENT][NAME_LENGTH], // first name
           lname[MAX_STUDENT][NAME_LENGTH]; // last name
      int  grades[MAX_STUDENT][NUM_HWS];        // homework grades
      int  nStu;                            // number of students in class

         // INPUT - read names and grades into arrays--------------------
      nStu = readData( title, fname, lname, grades );
      if ( nStu < 1 ) printf( "No data available\n" );
      else
         // OUTPUT - print report ---------------------------------------
         printReport( title, fname, lname, grades, nStu );

      return 0;
   }
//----------------------
// read data into arrays
//----------------------
   int readData( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] )
   {
      int n = 0; // game counter
      int h;     // HW counter
      FILE *inp;
      inp = fopen( "HW_Grades.txt", "r" );
      if ( inp == NULL )
         printf( "Open file error\n" );
      else
      {
         fgets( title, TITLE_LENGTH, inp );
         while ( fscanf( inp, "%s %s", &fname[n], &lname[n] ) !=EOF )  // read one row (name) at a time
         {
```

```
INPUT FILE -- HW_Grades.txt

COMP1200 Fall 2009 Homework Grades
Susan williams 100 100 98 95 97 98 95 95 95 36
Judith Holliday 100 100 100 97 100 99 100 100 100 93
Brenda Young 96 89 54 98 77 49 97 47 98 79
Edward Moore 100 100 100 100 100 99 100 100 93 100
Thomas Edwards 100 100 98 98 97 99 99 99 95 85
Robert Smyth 100 100 100 94 100 84 99 65 98 55
Carolyn Sims 100 100 88 98 100 84 100 85 100 65
David Simpson 100 65 25 50 97 65 80 10 30 65
Jefferson Dykes 100 100 100 74 97 89 100 100 100 85
Michael Brock 100 80 88 78 80 93 72 79 65 40
Mitchell Cleveland 100 80 100 95 95 99 100 50 80 85
```

grades_2D_cat.c

```c

            for ( h=0; h<NUM_HWS; h++ ) fscanf( inp, "%d", &grades[n][h] );

            n++;
         }
      }
      return n;
   }

//---------------------
// print the report
//---------------------
   void printReport( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
   {
      printHeader( title );
      printDetails( fname, lname, grades, nStu );
   }
//---------------------
// print body of report
//---------------------
   void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
   {
      int n,h;
      int HW_Totals[NUM_HWS] = {0};
      int stu_Total;
      char name[NAME_LENGTH*2];

      for ( n=0; n<nStu; n++ )
      {
         stu_Total = 0;
         strcpy( name, lname[n] );
         strcat( name, ", " );
         strcat( name, fname[n] );
         printf( "%-20s% ", name );
         for ( h=0; h<NUM_HWS; h++ )
         {
            printf( "%5d", grades[n][h] );
            stu_Total += grades[n][h]; // student's grades accumulator
            HW_Totals[h] += grades[n][h]; // HW accumulator
         }
         printf( " -- %4.1f\n", (double)stu_Total/NUM_HWS );
      }
      printf( "--------------------------------------------------------------------------------\n" );
         // print average by assignments
      printf( "Assignment Average   " );
      for ( h=0; h<NUM_HWS; h++ ) printf( "%5.1f", (double)HW_Totals[h]/nStu );
      printf( "\n" );
   }
//-------------------
// print headers
//-------------------
    void printHeader( char title[] )
   {
      printf( "                      %s", title );
      printf( "--------------------------------------------------------------------------------\n" );
      printf( "                       1    2    3    4    5    6    7    8    9   10     Ave\n" );
      printf( "--------------------------------------------------------------------------------\n" );
   }
```

```
                       COMP1200 Fall 2009 Homework Grades
--------------------------------------------------------------------------------
                        1    2    3    4    5    6    7    8    9   10      Ave
--------------------------------------------------------------------------------
williams, Susan       100  100   98   95   97   98   95   95   95   36 -- 90.9
Holliday, Judith      100  100  100   97  100   99  100  100  100   93 -- 98.9
Young, Brenda          96   89   54   98   77   49   97   47   98   79 -- 78.4
Moore, Edward         100  100  100  100  100   99  100  100   93  100 -- 99.2
Edwards, Thomas       100  100   98   98   97   99   99   99   95   85 -- 97.0
Smyth, Robert         100  100  100   94  100   84   99   65   98   55 -- 89.5
Sims, Carolyn         100  100   88   98  100   84  100   85  100   65 -- 92.0
Simpson, David        100   65   25   50   97   65   80   10   30   65 -- 58.7
Dykes, Jefferson      100  100  100   74   97   89  100  100  100   85 -- 94.5
Brock, Michael        100   80   88   78   80   93   72   79   65   40 -- 77.5
Cleveland, Mitchell   100   80  100   95   95   99  100   50   80   85 -- 88.4
--------------------------------------------------------------------------------
Assignment Average    99.6 92.2 86.5 88.8 94.5 87.1 94.7 75.5 86.7 71.6
```

grades_2D_cat.c

```
/* J Hundley - grades with strings
File name: grades_2D_outfile.c
Program: Students' names and HW grades
----------------------------------------------------------------------------
Read the names and grades into 2D arrays and print report.
----------------------------------------------------------------------------
*/
#include <stdio.h>
#include <string.h>
#define  MAX_STUDENT  20 // max number of students in class
#define  NUM_HWS      10 // number of assignments
#define  NAME_LENGTH  11 // number of characters in a name+1
#define  TITLE_LENGTH 41 // number of characters in the title+1

//-------------------
// function prototypes
//-------------------
   int  readData    ( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] ); //read data into arrays
   void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );

//-------------------
// main
//-------------------
    int main()
   {
      // declare and initialize variables
      char title[TITLE_LENGTH];             // file/report title
      char fname[MAX_STUDENT][NAME_LENGTH], // first name
           lname[MAX_STUDENT][NAME_LENGTH]; // last name
      int  grades[MAX_STUDENT][NUM_HWS];        // homework grades
      int  nStu;                            // number of students in class

         // INPUT - read names and grades into arrays--------------------
      nStu = readData( fname, lname, grades );
      if ( nStu < 1 ) printf( "No data available\n" );
      else
         // OUTPUT - print report ---------------------------------------
      printDetails( fname, lname, grades, nStu );

      return 0;
   }
//----------------------
// read data into arrays
//----------------------
    int readData( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] )
   {
      int n = 0;                // game counter
      int h;                    // HW counter
      char title[TITLE_LENGTH]; // holder for title, not used
      FILE *inp;
      inp = fopen( "HW_Grades.txt", "r" );
      if ( inp == NULL )
         printf( "Open file error\n" );
      else
      {
         fgets( title, TITLE_LENGTH, inp );
         while ( fscanf( inp, "%s %s", &fname[n], &lname[n] ) !=EOF )  // read one row (name) at a time
         {
            for ( h=0; h<NUM_HWS; h++ ) fscanf( inp, "%d", &grades[n][h] );
            n++;
         }
      }
      return n;
   }

//---------------------
// print body of report
//---------------------
    void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
   {
      int n,h;
      int HW_Totals[NUM_HWS] = {0};
      int stu_Total;
      char name[NAME_LENGTH*2];

      FILE *outfile;

      outfile = fopen( "courseGrades.txt", "w" );
      if ( outfile == NULL ) printf( "File not created" );
      else
      {
         fprintf( outfile, "%d\n", nStu );      // print number of records
         for ( n=0; n<nStu; n++ )
         {
            stu_Total = 0;
            strcpy( name, lname[n] );            // copy string
            strcat( name, ", " );                // concatenate two strings
            strcat( name, fname[n] );
            fprintf( outfile, "%-20s% ", name ); // print lname, fname
            for ( h=0; h<NUM_HWS; h++ )
            {
               stu_Total += grades[n][h];        // student's grades accumulator
            }
            fprintf( outfile, "%4.1f\n", (double)stu_Total/NUM_HWS ); // print average
         }
         fclose( outfile );                      // close file to est. EOF
      }
   }
/*
11
williams, Susan     90.9
Holliday, Judith    98.9
Young, Brenda       78.4
Moore, Edward       99.2
Edwards, Thomas     97.0
Smyth, Robert       89.5
Sims, Carolyn       92.0
Simpson, David      58.7
Dykes, Jefferson    94.5
Brock, Michael      77.5
Cleveland, Mitchell 88.4
*/
```

```c
// read names and grades
// sort by name and print
// Program: grades_2D_sort.c
// REVIEW: if-else-if and switch

#include <stdio.h>
#define NAME_LEN    20
#define MAX_STUDENT 20

   int    readData   ( char name[][NAME_LEN], double grade[] );
   void   printData( char name[][NAME_LEN], double grade[], int nStu );
   double get_QP( char ltr );
   char   get_ltrGrade( double numGrade );
   void   sort_by_name( char name[][NAME_LEN], double grade[], int nStu );

    int main()
   {
      char   name[MAX_STUDENT][NAME_LEN];
      double grade[MAX_STUDENT];
      int    nStu;

      nStu = readData( name, grade );
      if ( nStu > 0 ) printData( name, grade, nStu );

      return 0;
   }

    int  readData   ( char name[][NAME_LEN], double grade[] )
   {
      int n, nStu = 0;
      FILE *infile;
      infile = fopen( "courseGrades.txt", "r" );
      if ( infile == NULL ) printf( "File open error" );
      else
      {
         fscanf( infile, "%d ", &nStu );          // notice the space before using the fgets()
         for ( n=0; n<nStu; n++ )
         {
            fgets( name[n], NAME_LEN, infile );
            fscanf( infile, "%lf ", &grade[n] );  // notice the space before using the fgets()
         }
      }
      return nStu;
   }
```

```
Original Data
   williams, Susan      90.9
   Holliday, Judith     98.9
   Young, Brenda        78.4
   Moore, Edward        99.2
   Edwards, Thomas      97.0
   Smyth, Robert        89.5
   Sims, Carolyn        92.0
   Simpson, David       58.7
   Dykes, Jefferson     94.5
   Brock, Michael       77.5
   Cleveland, Mitchell  88.4

Sorted Data
   Brock, Michael       77.5
   Cleveland, Mitchell  88.4
   Dykes, Jefferson     94.5
   Edwards, Thomas      97.0
   Holliday, Judith     98.9
   Moore, Edward        99.2
   Simpson, David       58.7
   Sims, Carolyn        92.0
   Smyth, Robert        89.5
   Young, Brenda        78.4
   williams, Susan      90.9

Brock, Michael       77.5  C  2.0
Cleveland, Mitchell  88.4  B  3.0
Dykes, Jefferson     94.5  A  4.0
Edwards, Thomas      97.0  A  4.0
Holliday, Judith     98.9  A  4.0
Moore, Edward        99.2  A  4.0
Simpson, David       58.7  F  0.0
Sims, Carolyn        92.0  A  4.0
Smyth, Robert        89.5  B  3.0
Young, Brenda        78.4  C  2.0
williams, Susan      90.9  A  4.0
```

grades_2D_sort.c

```
 void printData( char name[][NAME_LEN], double grade[], int nStu )
{
   int n;
   char   ltrGrade[nStu];
   double qualPts[nStu];
   //----------------------------------------------------------------------
   // print original data
   printf( "Original Data\n" );
   for ( n=0; n<nStu; n++ ) printf( "  %-20s %3.1f \n", name[n],  grade[n] );

   sort_by_name( name, grade, nStu );
   // print sorted data
   printf( "\nSorted Data\n" );
   for ( n=0; n<nStu; n++ ) printf( "  %-20s %3.1f \n", name[n],  grade[n] );
   //----------------------------------------------------------------------
   printf( "\n" );

   for ( n=0; n<nStu; n++ )
   {
      ltrGrade[n] = get_ltrGrade( grade[n] );  // get letter grade for number grade
      qualPts[n]  = get_QP( ltrGrade[n] );     // get qual pts for letter grade
      printf( "%-20s %3.1f  %c  %3.1f\n", name[n],  grade[n], ltrGrade[n], qualPts[n] );
   }
}

 void sort_by_name( char name[][NAME_LEN], double grade[], int nStu )
{
   int    pass, nextMin, j;
   double hold_gr;
   char   hold_n[NAME_LEN];

   for (pass=0; pass<nStu-1; pass++)
   { // Exchange minimum with next array value.
      nextMin = pass;
      for (j=pass+1; j<nStu; j++)
      {
         if ( strcmp( name[j], name[nextMin] ) < 0 ) nextMin = j;
      }

      strcpy( hold_n,        name[nextMin] );  // swap names
      strcpy( name[nextMin], name[pass]    );
      strcpy( name[pass],    hold_n        );

      hold_gr = grade[nextMin];                // swap grades
      grade[nextMin] = grade[pass];
      grade[pass] = hold_gr;
   }
}
 char get_ltrGrade( double numGrade )
{
   char ltr;
   if      ( numGrade >= 90.0 ) ltr = 'A';
   else if ( numGrade >= 80.0 ) ltr = 'B';
   else if ( numGrade >= 70.0 ) ltr = 'C';
   else if ( numGrade >= 60.0 ) ltr = 'D';
   else                         ltr = 'F';
   return ltr;
}
```

```
 double get_QP( char ltr )
{
   double qp;
   switch (ltr)
   {
      case 'A': qp = 4.0;
         break;
      case 'B': qp = 3.0;
         break;
      case 'C': qp = 2.0;
         break;
      case 'D': qp = 1.0;
         break;
      default:  qp = 0.0;
         break;
   }
   return qp;
}
```

grades_2D_sort.c