

```

1 /* J Hundley - grades with strings
2 File name: grades_2D.c
3 Program: Students' names and HW grades
4 -----
5 Read the names and grades into 2D arrays and print report.
6 -----
7 */
8 #include <stdio.h>
9 #include <string.h>
10 #define MAX_STUDENT 20 // max number of students in class
11 #define NUM_HWS 10 // number of assignments
12 #define NAME_LENGTH 11 // number of characters in a name+1
13 #define TITLE_LENGTH 41 // number of characters in the title+1
14
15 //-----
16 // function prototypes
17 //-----
18 int readData ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] ); //read data into arrays
19 void printReport ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
20 void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
21 void printHeader ( char title[] ); //print headers
22
23 //-----
24 // main
25 //-----
26 int main()
27 {
28     // declare and initialize variables
29     char title[TITLE_LENGTH]; // file/report title
30     char fname[MAX_STUDENT][NAME_LENGTH], // first name
31           lname[MAX_STUDENT][NAME_LENGTH]; // last name
32     int grades[MAX_STUDENT][NUM_HWS]; // homework grades
33     int nStu; // number of students in class
34
35     // INPUT - read names and grades into arrays-----
36     nStu = readData( title, fname, lname, grades );
37     if ( nStu < 1 ) printf( "No data available\n" );
38     else
39         // OUTPUT - print report -----
40         printReport( title, fname, lname, grades, nStu );
41
42     return 0;
43 }
44 //-----
45 // read data into arrays
46 //-----
47 int readData( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] )
48 {
49     int n = 0; // game counter
50     int h; // HW counter
51     FILE *inp;
52     inp = fopen( "HW_Grades.txt", "r" );
53     if ( inp == NULL )
54         printf( "Open file error\n" );
55     else
56     {
57         fgets( title, TITLE_LENGTH, inp );
58         while ( fscanf( inp, "%s %s", &fname[n], &lname[n] ) != EOF ) // read one row (name) at a time
59             n++;
60     }

```

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.c

```

60
61         for ( h=0; h<NUM_HWS; h++ ) fscanf( inp, "%d", &grades[n][h] );
62
63         n++;
64     }
65 }
66 return n;
67 }
68
69 //-----
70 // print the report
71 //-----
72 void printReport( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
73 {
74     printHeader( title );
75     printDetails( fname, lname, grades, nStu );
76 }
77 //-----
78 // print body of report
79 //-----
80 void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
81 {
82     int n,h;
83     int HW_Totals[NUM_HWS] = {0};
84     int stu_Total;
85
86     for ( n=0; n<nStu; n++ )
87     {
88         stu_Total = 0;
89         printf( "%-12s %-12s ", fname[n], lname[n] );
90         for ( h=0; h<NUM_HWS; h++ )
91         {
92             printf( "%5d", grades[n][h] );
93             stu_Total += grades[n][h]; // student's grades accumulator
94             HW_Totals[h] += grades[n][h]; // HW accumulator
95         }
96         printf( " -- %4.1f\n", (double)stu_Total/NUM_HWS );
97     }
98
99     printf( "-----\n" );
100
101     // print average by assignments
102     printf( "Assignment Average      " );
103     for ( h=0; h<NUM_HWS; h++ ) printf( "%5.1f", (double)HW_Totals[h]/nStu );
104     printf( "\n" );
105 }
106
107 //-----
108 // print headers
109 //-----
110 void printHeader( char title[] )
111 {
112     printf( "          %s", title );
113     printf( "-----\n" );
114     printf( "          1    2    3    4    5    6    7    8    9    10    Ave\n" );
115     printf( "-----\n" );
116 }
117

```

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	

```

1 /* J Hundley - grades with strings
2 File name: grades_2D_cat.c
3 Program: Students' names and HW grades
4 -----
5 Read the names and grades into 2D arrays and print report.
6 -----
7 */
8 #include <stdio.h>
9 #include <string.h>
10 #define MAX_STUDENT 20 // max number of students in class
11 #define NUM_HWS 10 // number of assignments
12 #define NAME_LENGTH 11 // number of characters in a name+1
13 #define TITLE_LENGTH 41 // number of characters in the title+1
14
15 //-----
16 // function prototypes
17 //-----
18 int readData ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] ); //read data into arrays
19 void printReport ( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
20 void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu );
21 void printHeader ( char title[] ); //print headers
22
23 //-----
24 // main
25 //-----
26 int main()
27 {
28     // declare and initialize variables
29     char title[TITLE_LENGTH]; // file/report title
30     char fname[MAX_STUDENT][NAME_LENGTH], // first name
31           lname[MAX_STUDENT][NAME_LENGTH]; // last name
32     int grades[MAX_STUDENT][NUM_HWS]; // homework grades
33     int nStu; // number of students in class
34
35     // INPUT - read names and grades into arrays-----
36     nStu = readData( title, fname, lname, grades );
37     if ( nStu < 1 ) printf( "No data available\n" );
38     else
39         // OUTPUT - print report -----
40         printReport( title, fname, lname, grades, nStu );
41
42     return 0;
43 }
44 //-----
45 // read data into arrays
46 //-----
47 int readData( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS] )
48 {
49     int n = 0; // game counter
50     int h; // HW counter
51     FILE *inp;
52     inp = fopen( "HW_Grades.txt", "r" );
53     if ( inp == NULL )
54         printf( "Open file error\n" );
55     else
56     {
57         fgets( title, TITLE_LENGTH, inp );
58         while ( fscanf( inp, "%s %s", &fname[n], &lname[n] ) != EOF ) // read one row (name) at a time
59             {

```

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

```

60
61         for ( h=0; h<NUM_HWS; h++ ) fscanf( inp, "%d", &grades[n][h] );
62
63         n++;
64     }
65 }
66 return n;
67 }
68
69 //-----
70 // print the report
71 //-----
72 void printReport( char title[], char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
73 {
74     printHeader( title );
75     printDetails( fname, lname, grades, nStu );
76 }
77 //-----
78 // print body of report
79 //-----
80 void printDetails( char fname[][NAME_LENGTH], char lname[][NAME_LENGTH], int grades[][NUM_HWS], int nStu )
81 {
82     int n,h;
83     int HW_Totals[NUM_HWS] = {0};
84     int stu_Total;
85     char name[NAME_LENGTH*2];
86
87     for ( n=0; n<nStu; n++ )
88     {
89         stu_Total = 0;
90         strcpy( name, lname[n] );
91         strcat( name, ", " );
92         strcat( name, fname[n] );
93         printf( "%-20s", name );
94         for ( h=0; h<NUM_HWS; h++ )
95         {
96             printf( "%5d", grades[n][h] );
97             stu_Total += grades[n][h]; // student's grades accumulator
98             HW_Totals[h] += grades[n][h]; // HW accumulator
99         }
100         printf( " -- %4.1f\n", (double)stu_Total/NUM_HWS );
101     }
102     printf( "-----\n" );
103     // print average by assignments
104     printf( "Assignment Average  " );
105     for ( h=0; h<NUM_HWS; h++ ) printf( "%5.1f", (double)HW_Totals[h]/nStu );
106     printf( "\n" );
107 }
108
109 //-----
110 // print headers
111 //-----
112 void printHeader( char title[] )
113 {
114     printf( "          %s", title );
115     printf( "-----\n" );
116     printf( "          1    2    3    4    5    6    7    8    9    10    Ave\n" );
117     printf( "-----\n" );
118 }
119
120 }

```

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	

```

/* 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
*/

```

```

1 // read names and grades
2 // sort by name and print
3 // Program: grades_2D_sort.c
4 // REVIEW: if-else-if and switch
5
6 #include <stdio.h>
7 #define NAME_LEN 20
8 #define MAX_STUDENT 20
9
10 int readData ( char name[][NAME_LEN], double grade[] );
11 void printData( char name[][NAME_LEN], double grade[], int nStu );
12 double get_QP( char ltr );
13 char get_ltrGrade( double numGrade );
14 void sort_by_name( char name[][NAME_LEN], double grade[], int nStu );
15
16 int main()
17 {
18     char name[MAX_STUDENT][NAME_LEN];
19     double grade[MAX_STUDENT];
20     int nStu;
21
22     nStu = readData( name, grade );
23     if ( nStu > 0 ) printData( name, grade, nStu );
24
25     return 0;
26 }
27
28 int readData ( char name[][NAME_LEN], double grade[] )
29 {
30     int n, nStu = 0;
31     FILE *infile;
32     infile = fopen( "courseGrades.txt", "r" );
33     if ( infile == NULL ) printf( "File open error" );
34     else
35     {
36         fscanf( infile, "%d ", &nStu ); // notice the space before using the fgets()
37         for ( n=0; n<nStu; n++ )
38         {
39             fgets( name[n], NAME_LEN, infile );
40             fscanf( infile, "%lf ", &grade[n] ); // notice the space before using the fgets()
41         }
42     }
43     return nStu;
44 }
45

```

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

```

46 void printData( char name[][NAME_LEN], double grade[], int nStu )
47 {
48     int n;
49     char ltrGrade[nStu];
50     double qualPts[nStu];
51     //-----
52     // print original data
53     printf( "Original Data\n" );
54     for ( n=0; n<nStu; n++ ) printf( " %-20s %3.1f \n", name[n], grade[n] );
55
56     sort_by_name( name, grade, nStu );
57     // print sorted data
58     printf( "\nSorted Data\n" );
59     for ( n=0; n<nStu; n++ ) printf( " %-20s %3.1f \n", name[n], grade[n] );
60     //-----
61     printf( "\n" );
62
63     for ( n=0; n<nStu; n++ )
64     {
65         ltrGrade[n] = get_ltrGrade( grade[n] ); // get letter grade for number grade
66         qualPts[n] = get_QP( ltrGrade[n] ); // get qual pts for letter grade
67         printf( "%-20s %3.1f %c %3.1f\n", name[n], grade[n], ltrGrade[n], qualPts[n] );
68     }
69 }
70
71
72 void sort_by_name( char name[][NAME_LEN], double grade[], int nStu )
73 {
74     int pass, nextMin, j;
75     double hold_gr;
76     char hold_n[NAME_LEN];
77
78     for (pass=0; pass<nStu-1; pass++)
79     { // Exchange minimum with next array value.
80         nextMin = pass;
81         for (j=pass+1; j<nStu; j++)
82         {
83             if ( strcmp( name[j], name[nextMin] ) < 0 ) nextMin = j;
84         }
85
86         strcpy( hold_n, name[nextMin] ); // swap names
87         strcpy( name[nextMin], name[pass] );
88         strcpy( name[pass], hold_n );
89
90         hold_gr = grade[nextMin]; // swap grades
91         grade[nextMin] = grade[pass];
92         grade[pass] = hold_gr;
93     }
94 }
95
96 char get_ltrGrade( double numGrade )
97 {
98     char ltr;
99     if ( numGrade >= 90.0 ) ltr = 'A';
100    else if ( numGrade >= 80.0 ) ltr = 'B';
101    else if ( numGrade >= 70.0 ) ltr = 'C';
102    else if ( numGrade >= 60.0 ) ltr = 'D';
103    else ltr = 'F';
104    return ltr;
105 }
106

```

```

110 double get_QP( char ltr )
111 {
112     double qp;
113     switch (ltr)
114     {
115         case 'A': qp = 4.0;
116             break;
117         case 'B': qp = 3.0;
118             break;
119         case 'C': qp = 2.0;
120             break;
121         case 'D': qp = 1.0;
122             break;
123         default: qp = 0.0;
124             break;
125     }
126     return qp;
127 }

```

grades\_2D\_sort.c