```c
#include <stdio.h>
#include <string.h>

   int  maxIndex  (double x[][3], int numElem, int col);
   void printData (char name[][10], double grade[][3], int numStu);
   void sortData  (char name[][10], double grade[][3], int numStu, int col);
   void swapDouble(double *d1, double *d2);
   void swapString(char s1[], char s2[], int col);

    int main()
   {
      double grade[10][3], ave[10];
      int ns=0, numStu, iMax;
      char name[10][10];

      FILE *infile;
      infile = fopen("grades.txt","r");
      while(fscanf(infile,"%s %lf %lf",&name[ns],&grade[ns][0],&grade[ns][1]) != EOF)
      {
         grade[ns][2] = (grade[ns][0] + grade[ns][1]) / 2.0;
         ns++;
      }
      numStu = ns;                          // number of students
      iMax = maxIndex(grade,numStu,2);  // find the index of max average
      printf("%s has the highest average: %.1f\n\n",name[iMax],grade[iMax][2]);

      printf("INPUT DATA:\n");
      printData(name,grade,numStu);
      sortData(name,grade,numStu,2);
      printf("\nSORTED DATA:\n");
      printData(name,grade,numStu);

      return  0;
   }

// return the index of the maximum values in an array
    int maxIndex(double x[][3], int numElem, int col)
   {
      int k, max_k;
      double max_x;
      max_x = x[0][col];          // Determine maximum value in the array.
      for (k=1; k<numElem; k++)
      {
         if (x[k][col] > max_x)
         {
            max_x = x[k][col];
            max_k = k;
         }
      }
      return max_k;
   }

// print data in table
    void printData(char name[][10], double grade[][3], int numElem)
   {
```

```
56       int r, c;
57       for (r=0;r<numElem;r++)
58       {
59          printf("%-10s", name[r]);
60          for (c=0;c<3;c++)
61             printf("   %.1lf", grade[r][c]);
62          printf("\n");
63       }
64    }
65
66 // sort  parallel arrays
67     void sortData(char name[][10], double grade[][3], int numStu, int col)
68    {
69       int pass, nextMin, j, i;
70       double hold;
71       for (pass=0; pass<numStu-1; pass++)
72       {   /* Exchange minimum with next array value.  */
73          nextMin = pass;
74          for (j=pass+1; j<numStu; j++)
75             if (grade[j][col] < grade[nextMin][col])
76                nextMin = j;
77
78          swapString(name[pass], name[nextMin], 10);
79          for (i=0;i<3;i++)
80             swapDouble(&grade[pass][i],&grade[nextMin][i]);
81       }
82    }
83
84 // swap two doubles
85     void swapDouble(double *d1, double *d2)
86    {
87       double hold;
88       hold = *d1;
89       *d1 = *d2;
90       *d2 = hold;
91    }
92
93 // swap twp strings
94     void swapString(char s1[], char s2[], int col)
95    {
96       char hold[col];
97       strcpy(hold,s1);
98       strcpy(s1,s2);
99       strcpy(s2,hold);
100   }
```

```
Vance has the highest average: 92.5

INPUT DATA:
Adams         87.0    79.0    83.0
Dawson        90.0    87.0    88.5
Floyd         67.0    66.0    66.5
Noble         59.0    75.0    67.0
Sanders       89.0    76.0    82.5
Trace         67.0    98.0    82.5
Vance         98.0    87.0    92.5
Willis        77.0    88.0    82.5
Yates         99.0    71.0    85.0

SORTED DATA:
Floyd         67.0    66.0    66.5
Noble         59.0    75.0    67.0
Sanders       89.0    76.0    82.5
Trace         67.0    98.0    82.5
Willis        77.0    88.0    82.5
Adams         87.0    79.0    83.0
Yates         99.0    71.0    85.0
Dawson        90.0    87.0    88.5
Vance         98.0    87.0    92.5
```