University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 160 — Submit name: `stutestnames.cpp`
Project 12 — Due: see http://ece160.org

Write a program to read a file containing student names and associated test scores. The data file will be named "`stutestnames.txt`". The first line of the file will contain two integers each one occupying four columns. This first integer (in columns 1-4) will be the number of students. The second integer (in columns 5-8) will be the number of exams for each student. A sample file is below. Note this is a sample file – the names, scores, number of scores, and number of lines may change; The formatting of the data, however, will remain constant. That is the name will be in columns 1-20; each test score will occupy four columns.

```
..../....1..../....2..../....3..../..  ← Column number (not in file)
   6   4
Scott,Montgomery      80  90  92  68
Spock,Mr.            100 100 100 100
Sulu,Hikiru           80  85  90  95
Checkov,Pavel        100  95  90  90
Kirk,James           100  95 100  98
McCoy,Leonard         92  87  85  90
```

The output should be a nicely formatted table (with headings) showing each students name, their test scores, and their averages. The average for each column (including the "Average" column) must be shown as well. All averages should be displayed to two decimal places. If the data file above was used, the output should look as follows:

```
Name                    Test 1  Test 2  Test 3  Test 4  Average
Scott,Montgomery          80      90      92      68     82.50
Spock,Mr.                100     100     100     100    100.00
Sulu,Hikiru               80      85      90      95     87.50
Checkov,Pavel            100      95      90      90     93.75
Kirk,James               100      95     100      98     98.25
McCoy,Leonard             92      87      85      90     88.50
Averages                  92.00   92.00   92.83   90.17  91.75
```

Your program must have (and make appropriate use of) three symbols, namely, `MAXSTU`, `MAXTST`, and `MAXCHR` (see sample code)

Additional notes and specifications:

- Your program must use a two dimensional array for the test scores.
- Although the sample file has six students and four tests, your program should be able to handle up to MAXSTU students, and MAXTST tests.
- In the interest of forcing you to use arrays, you must read the entire file and close it prior to outputting anything.
- You do not have to worry about "wrapping" of lines if there are more tests than will fit on one line.
- The filename specified in the `fopen()` call may not contain any path; it must simply be `"stutestnames.txt".`
- EXTRA CREDIT – for 50 additional points, sort the output by name. Obviously, whenever you make a swap, you need to swap the name, all associated grades, and student average. See the function `strcmp()` for more info about comparing strings.
- EXTRA EXTRA CREDIT – for an additional 50 (total of 100), don't swap any data – use an index array to point to the data to print. You will be using a subscripted variable as a subscript.

Some useful code:

```
#define _CRT_SECURE_NO_DEPRECATE
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MAXSTU 100
#define MAXTST 40
#define MAXCHR 20

void main()
{
      char name[MAXSTU][MAXCHR+1];
      char linbuf[100];
      int T[MAXSTU][MAXTST];
      int numstu,numtst,i,j;

      FILE *f;

      if ((f=fopen("stutestnames.txt","rt"))==NULL)
      {
            printf("Error opening file...");
            exit(0);
      }
      fgets(linbuf,100,f);
      numstu=atoi(linbuf);
      numtst=atoi(&linbuf[4]);

      for (i=0; i<numstu; i++)
      {
            fgets(linbuf,100,f);
            strncpy(name[i],linbuf,MAXCHR);
            name[i][MAXCHR]='\0';
            for (j=0;j<numtst;j++)
                  T[i][j]=atoi(&linbuf[MAXCHR+j*4]);
      }
      fclose(f);

      for (i=0; i<numstu; i++)
      {
            printf("%-20s",name[i]);
            for (j=0;j<numtst;j++)
                  printf("%4d",T[i][j]);
            printf("\n");
      }
}
```