```c
// J Hundley
// assign09
// April 16, 2015
/* 2015 AU softball games results
read file into two char arrays (date, opponent),
   a 2D integer array (AU runs, opp runs, innings, attendance) and
  a double 1D array (game time computed from hours and minutes).
*/
#include <stdio.h>
//*****CONSTANT*****
#define FILENAME   "2015_AU_softball_SEC.txt"
#define MAXGAMES    25  // estimated number of SEC games in season
#define NUMINNINGS  7   // number of innings in a game
#define NUMCOLS     4   // number of columns in game info array
#define AU_COL      0   // Auburn runs in 1st column of game info array
#define OPP_COL     1   // Opp runs in 2nd column of game info array
#define INN_COL     2   // no. of innings in 3rd column of game info array
#define ATT_COL     3   // attendance in 4th column of game info array
#define MIN_IN_HOUR 60  // number of minutes in an hour
#define TITLE_LEN   27  // title length
#define DATE_LEN    12  // max length of date
#define TEAM_LEN    15  // max length of opp team name

// FUNCTION PROTOTYPES=================================================================
int    getFileData ( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                     int gameStats[][NUMCOLS], double hours[] );
void   printSummary( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                     int gameStats[][NUMCOLS], double hours[], int numGames );
double intColMean( int x[][NUMCOLS], int n, int colNum );
double doubleMean( double x[], int n );
// MAIN ================================================================================
int main() {
   char   title[TITLE_LEN], date[MAXGAMES][DATE_LEN], team[MAXGAMES][TEAM_LEN];
   int    gameInfo[MAXGAMES][NUMCOLS];
   double gameLength[MAXGAMES];
   int    numGames = 0;
   // INPUT
   numGames = getFileData( title, date, team, gameInfo, gameLength );
   if (numGames > 0) {
      // OUTPUT
      printSummary( title, date, team, gameInfo, gameLength, numGames );
   }
   else {
      printf("No data read. Program ending.");
   }
   return 0;
}
// FUNCTION DEFINITIONS================================================================
// read file. store selected data in 1D integer arrays ================================
int getFileData ( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                  int gameStats[][NUMCOLS], double hours[] ) {
   int g = 0,      // counter
       hour, min;  // game time
   FILE * filePtr; // file pointer
//*****INPUT*****
//open input data file
   filePtr = fopen(FILENAME,"r");
// check for good file open
   if (filePtr == NULL)
      printf("File open error.\n");
   else { // good file open continue program
      fgets( title, TITLE_LEN, filePtr );
          // 14-Mar-2015 Ole_Miss 14 3 5 440 2 75
      while( fscanf( filePtr, " %s %s %d %d %d %d %d %d",
                     date[g], team[g], &gameStats[g][AU_COL], &gameStats[g][OPP_COL],
                     &gameStats[g][INN_COL],&gameStats[g][ATT_COL],&hour,&min ) != EOF ) {
         hours[g] = hour + (double)min/MIN_IN_HOUR;
         g++;
      } // while
   } // else
   return g;
}
```

```
// print statistics summary report ========================================================
void   printSummary( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                     int gameStats[][NUMCOLS], double hours[], int numGames ) {
   int    g;
   double meanAtt, meanTime;

   // find max attendance and game length
   meanAtt  = intColMean( gameStats, numGames, ATT_COL );
   meanTime = doubleMean( hours, numGames );

   // report title, headings
   printf( "      2015 AU Softall Summary as of %s\n", date[numGames-1] );
   printf( " Game date   Opposing team   W/L Score Inns Attend Time(hours)  \n" );
   // game information
   for ( g=0; g<numGames; g++ ){
      printf( "%s  %-15s  ", date[g], team[g] );
        // Win or Loss
      if ( gameStats[g][AU_COL] > gameStats[g][OPP_COL] )
         printf( "W" );
      else
         printf( "L" );
        // score, innings, attendance
      printf( "  %02d-%02d  %2d   %4d",
            gameStats[g][AU_COL], gameStats[g][OPP_COL],
            gameStats[g][INN_COL],gameStats[g][ATT_COL] );
       // above average attendance
      if ( gameStats[g][ATT_COL] > meanAtt )
         printf( "*  " );
      else
         printf( "   " );
       // length of game
      printf( "%4.2f", hours[g] );
       // longer than average game
      if ( hours[g] > meanTime )
         printf( "** \n" );
      else
         printf( "\n" );
   } // end for each game
   // print average attendance and time in footer
   printf( " * attendance above average:       %6.2f\n", meanAtt );
   printf( "** games length longer than average: %4.2f\n", meanTime );


} // end printSummary
// return the average value of a column of array x with n rows ============================
double intColMean ( int x[][NUMCOLS], int n, int colNum ) {
   int k;
   int sum=0;
   for ( k=0; k<n; k++ ) {
      sum += x[k][colNum];
   }
   return (double)sum/n;
}
// return the average value of the array x with n elements ================================
double doubleMean ( double x[], int n ) {
   int    k;
   double sum=0;
   for ( k=0; k<n; k++ ) {
      sum += x[k];
   }
   return sum/n;
}
```

Read all instructions before beginning your work.

COMP1200-C - Assign 10
Due midnight – Thursday – April 16, 2015
Submit `assign10.c` via Canvas

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points for not doing so.]

***Before you start writing your program**:*

Read these instructions carefully. A development plan is a process that guides you through solving a problem and creating an algorithm. Download the `2015_AU_softball_SEC.txt` data file from the assign10 announcement and save in your COMP1200/assign10 folder. assign10.m will look in the folder where it is saved for the data file. A development plan is a process that guides you through solving a problem and creating an algorithm. Create your own algorithm and use it as comments throughout your program. Use section comments to group your statements as well as comments from your algorithm.

***Problem*:**

***Program*: assign10.c**

The 2015 Auburn softball team is having a great season. In this assignment, you will read some game result from `2015_AU_softball_SEC.txt` and print a report shown below. The season is not over; you do not know how many games are in the file. The following information is in `2015_AU_softball_SEC.txt` for each SEC game of the season:

```
2015 AU Softball SEC Games
14-Mar-2015 Ole_Miss 14 3 5 440 2 7    <= date, opponent, AU runs, opp runs, innings, attendance, hours, minutes
15-Mar-2015 Ole_Miss 7 3 7 200 2 18
16-Mar-2015 Ole_Miss 10 2 6 171 2 25
. . .
```

In assign10.c, you will use two char arrays (date, opponent), a 2D integer array (AU runs, opp runs, innings, attendance) and a double 1D array (game time computed from hours and minutes).

Use the following user-defined functions to perform the described tasks. The program structure diagram provides a guide to the relationship of the functions and the information passed to and from the functions.

```
// FUNCTION PROTOTYPES============================================================================
int getFileData ( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                  int gameStats[][NUMCOLS], double hours[] );
```

Function opens and read the data file. If there is an open error, print a message and return to main. Read all data and save specified data in 2D and 1D arrays as described above. Return the number of games read. If there is a file open error, the returned number of games counter will be zero. Use this information in main to determine if the program should be continued. If no games were read, print a message and do not continue the program.

```
void printSummary( char title[], char date[][DATE_LEN], char team[][TEAM_LEN],
                   int gameStats[][NUMCOLS], double hours[], int numGames );
```

Function uses the data in the arrays and the number of games to print a summary report. See more detained information below.

You will also need functions that find and return the average of a column in a 2D integer array and a 1D double array.

***Problem Constants*:**

```
FILENAME   "2015_AU_softball_SEC.txt"
MAXGAMES    25  // estimated number of SEC games in season
NUMINNINGS  7   // number of innings in a game
NUMCOLS     4   // number of columns in game info array
AU_COL      0   // Auburn runs in 1st column of game info array
OPP_COL     1   // Opp runs in 2nd column of game info array
INN_COL     2   // no. of innings in 3rd column of game info array
ATT_COL     3   // attendance in 4th column of game info array
MIN_IN_HOUR 60  // number of minutes in an hour
TITLE_LEN   27  // title length
DATE_LEN    12  // max length of date
TEAM_LEN    15  // max length of opp team name
```

FINAL EXAM
Wed, Apr 29
during class

***Problem Inputs*:**

See above.

***Problem Outputs*:**

See above.

***Other variables*:**

As needed.

New commands:
`fgets()`
left align, zero fill
2D char arrays
2D indexing

Instructions:

- ☐ See Standards for Documentation of C Programs on the Resources page on Canvas.
- ☐ Insert comments at the top and throughout each file.
  - Include the follow comments at the beginning of this (and ALL) files.
    - // submitter's name, **GROUP #** — **Grade of ZERO for files with submitter name not part of Canvas group**
    - // other group members' names — **Type "none" if submitting alone.**
    - // assignment number — **Zero points for comments if no collaboration statement**
    - // date you completed the assignment
    - **// statement(s) about collaboration**
    - // a short narrative about what the file does
  - Use the algorithm given as comments throughout your program.
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Use **Generate CSD** to ensure correct indenting.
- ☐ Using constants defined at the top of your program allows flexibility when information is changed in the future.
  - Represent ALL given values as constants.
  - Other than zero and optional one for counting initialization, **use the given constant names NOT numbers for the information that they represent**.

-5 points for absence of any of these required comments at the top at the top of each file.

If you do not submit individually, there will be a 5 POINTS PENALTY for not joining a group. Groups can be 2-4 students. DO NOT join a group unless you have worked with the other members. If you do, you will be removed from the group and given the grade of zero.

Sample Input/Output:

`2015_AU_softball_SEC.txt`

```
2015 AU Softball SEC Games
14-Mar-2015 Ole_Miss 14 3 5 440 2 7
15-Mar-2015 Ole_Miss 7 3 7 200 2 18
16-Mar-2015 Ole_Miss 10       2 6 171 2 25
20-Mar-2015 #12_KENTUCKY 5 0 7 1033 2 24
21-Mar-2015 #12_KENTUCKY 4 1 7 1300 2 21
21-Mar-2015 #12_KENTUCKY 9 7 7 1418 2 8
27-Mar-2015 #20_Missouri 14 6 5 558 2 35
28-Mar-2015 #20_Missouri 4 5 8 518 3 0
29-Mar-2015 #20_Missouri 10 8 7 543 3 15
02-Apr-2015 #11_TENNESSEE 6 2 7 1316 2 17
03-Apr-2015 #11_TENNESSEE 6 8 7 1736 2 52
04-Apr-2015 #11_TENNESSEE 12 9 7 1796 2 57
10-Apr-2015 #23_Texas_A&M 8 1 7 902 2 39
11-Apr-2015 #23_Texas_A&M 4 12 5 930 2 15
11-Apr-2015 #23_Texas_A&M 12 4 7 930 2 13
```

`fgets()`. Use in report title.

game data: date, opponent, AU runs, opp runs, innings, attendance, hours, minutes

```
      2015 AU Softall Summary as of 11-Apr-2015
 Game date   Opposing team   W/L Score Inns Attend Time(hours)
14-Mar-2015  Ole_Miss         W  14-03   5    440   2.12
15-Mar-2015  Ole_Miss         W  07-03   7    200   2.30
16-Mar-2015  Ole_Miss         W  10-02   6    171   2.42
20-Mar-2015  #12_KENTUCKY     W  05-00   7   1033*  2.40
21-Mar-2015  #12_KENTUCKY     W  04-01   7   1300*  2.35
21-Mar-2015  #12_KENTUCKY     W  09-07   7   1418*  2.13
27-Mar-2015  #20_Missouri     W  14-06   5    558   2.58**
28-Mar-2015  #20_Missouri     L  04-05   8    518   3.00**
29-Mar-2015  #20_Missouri     W  10-08   7    543   3.25**
02-Apr-2015  #11_TENNESSEE    W  06-02   7   1316*  2.28
03-Apr-2015  #11_TENNESSEE    L  06-08   7   1736*  2.87**
04-Apr-2015  #11_TENNESSEE    W  12-09   7   1796*  2.95**
10-Apr-2015  #23_Texas_A&M    W  08-01   7    902   2.65**
11-Apr-2015  #23_Texas_A&M    L  04-12   5    930*  2.25
11-Apr-2015  #23_Texas_A&M    W  12-04   7    930*  2.22
 * attendance above average:       919.40
** games length longer than average: 2.52
```

The first part of the title is the first line in the data file. The date is the date of the last game.

String columns are left aligned.
Numeric columns are right aligned.
Scores are zero filled.

Mark the attendance > average attendance with *.
Mark the time > average time with **.

Add a footer key with the average attendance and game length.

***Submit via* Canvas:**

`assign10.c` C program file