```c
// J Hundley
// assign09
// April 16, 2015
/* 2015 AU softball games results
read file but save selected data - 2D int arrays
max/mean/min AU runs scored: all, SEC with game #
max/mean/min runs allowed:   all, SEC with game #
max/mean/min point spread:   all, SEC with game #
game(s) with extra innings; regular game 7 innings
length of longest game
*/
#include <stdio.h>
//*****CONSTANT*****
#define FILENAME   "2015_AU_softball09.txt"
#define MAXGAMES    50  // estimated number of games in season
#define NUMSEC      25  // estimated number of SEC games in season
#define SECCODE     '*' // code for SEC games
#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

// FUNCTION PROTOTYPES ============================================================================
int    getFileData ( char code[], int gameStats[][NUMCOLS], double hours[] );
void   printSummary( char code[], int gameStats[][NUMCOLS], double hours[], int numGames );
int    getSecGames ( char code[], int arrayAll[][NUMCOLS], int numGames, int arraySec[][NUMCOLS] );
int    extraInnings( int gameStats[][NUMCOLS], int numGames, int extra[] );
int    intColMax ( int x[][NUMCOLS], int n, int colNum );
int    intColMin ( int x[][NUMCOLS], int n, int colNum );
double intColMean( int x[][NUMCOLS], int n, int colNum );
double doubleMax ( double x[], int n );
// MAIN PROTOTYPES ================================================================================
int main() {
   char   code[MAXGAMES];
   int    gameInfo[MAXGAMES][NUMCOLS];
   double gameLength[MAXGAMES];
   int    numGames = 0;
   // INPUT
   numGames = getFileData( code, gameInfo, gameLength );
   if (numGames > 0) {
      // OUTPUT
      printSummary( code, 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 code[], int gameStats[][NUMCOLS], double hours[] ) {
   int g = 0,      // counter
       hour, min,  // game time
       iHold;      // hold integers not saved
   char cHold;     // hold characters not saved
   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
```

```
          // x  15-13-1  5-5-1  5  845  2:15
      while( fscanf( filePtr, " %c %d%c%d%c%d %d%c%d%c%d %d %d %d%c%d",
         &code[g], &gameStats[g][AU_COL],&cHold,&iHold,&cHold,&iHold,      // Auburn stats
                   &gameStats[g][OPP_COL],&cHold,&iHold,&cHold,&iHold,     // opp stats
                   &gameStats[g][INN_COL],&gameStats[g][ATT_COL],&hour,&cHold,&min ) != EOF ) {
         hours[g] = hour + (double)min/MIN_IN_HOUR;
         g++;
      } // while
   } // else
   return g;
}
// print statistics summary report ============================================================
void printSummary( char code[], int gameStats[][NUMCOLS], double hours[], int numGames ) {
   int    g;
   int    arraySec[NUMSEC][NUMCOLS],
          extra[MAXGAMES];
   int    numSecGames, numExtra;
   int    minRuns, maxRuns;
   double aveRuns;

   printf( "      2015 AU Softall Summary\n" );
   printf( "                #games Min Mean Max\n" );

// all games runs
// max/mean/min AU runs scored: all, with # of games
   printf( "Runs scored-all   %2d    %2d  %4.1f %2d\n",
        numGames,intColMin(gameStats,numGames,AU_COL),
        intColMean(gameStats,numGames,AU_COL),intColMax(gameStats,numGames,AU_COL) );
// max/mean/min runs allowed:   all
   printf( "Runs allowed-all        %2d  %4.1f %2d\n",
        intColMin(gameStats,numGames,OPP_COL),
        intColMean(gameStats,numGames,OPP_COL),intColMax(gameStats,numGames,OPP_COL) );

// SEC games runs
// get the Auburn runs for the SEC game
   numSecGames = getSecGames( code, gameStats, numGames, arraySec );
// max/mean/min AU runs scored: SEC, with # of games
   printf( "Runs scored-SEC   %2d    %2d  %4.1f %2d\n",
        numSecGames,intColMin(arraySec,numSecGames,AU_COL),
        intColMean(arraySec,numSecGames,AU_COL), intColMax(arraySec,numSecGames,AU_COL) );

 // max/mean/min runs allowed:   SEC
   printf( "Runs allowed-SEC        %2d  %4.1f %2d\n",
        intColMin(arraySec,numSecGames,OPP_COL),
        intColMean(arraySec,numSecGames,OPP_COL),intColMax(arraySec,numSecGames,OPP_COL) );

// game(s) with extra innings; regular game 7 innings
   numExtra = extraInnings( gameStats, numGames, extra );
   printf( "\nGames with extra innings:\n");
   if (numExtra > 0 ) {
      for ( g=0; g<numExtra; g++ ) {
         printf( " %d", extra[g] );
      } // end for g
      printf( "\n" );

   } // if numExtra
   else {
      printf( "none\n" );

   } // end else
   // longest game for all and SEC
   printf( "\nLongest game played: %.2f hours\n", doubleMax( hours, numGames ) );

} // end printSummary
```

```
// copy information for SEC games to another array ================================================
int getSecGames( char code[], int arrayAll[][NUMCOLS], int numGames, int arraySec[][NUMCOLS] ) {
   int g, c, count = 0;
   for ( g=0; g<numGames; g++ ) {
      if( code[g] == SECCODE ) {
         for ( c=0; c<NUMCOLS; c++ ) {
            arraySec[count][c] = arrayAll[g][c];
         } // end for c
         count ++;
      } // end if code
   } // end for g
   return count;
}
// return the maximum value in the array x with n elements ========================================
int intColMax ( int x[][NUMCOLS], int n, int colNum ) {
   int k;
   double max_x;
   max_x = x[0][colNum];
   for ( k=1; k<n; k++ ) {
      if ( x[k][colNum] > max_x ) {
         max_x = x[k][colNum];
      } // end if x
   } // for k
   return max_x;
}
// return the minimum value in the array x with n elements ========================================
int intColMin ( int x[][NUMCOLS], int n, int colNum ) {
   int k;
   double min_x;
   min_x = x[0][colNum];
   for ( k=1; k<n; k++ ) {
      if ( x[k][colNum] < min_x ) {
         min_x = x[k][colNum];
      }
   }
   return min_x;
}
// return the average value of the array x with n elements ========================================
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 a list of the games with extra innings ==================================================
int extraInnings( int arrayAll[][NUMCOLS], int numGames, int extra[] ) {
   int g, count=0;
   for ( g=0; g<numGames; g++ ) {
      if ( arrayAll[g][INN_COL] > NUMINNINGS ) {
         extra[count] = g+1;
         count++;
      } // end if arrayAll
   } // end g
   return count;
}
// return the maximum value in the array x with n elements ========================================
double doubleMax( double x[], int n ) {
  int k;
   double max_x;
   max_x = x[0];
   for ( k=1; k<n; k++ )
   {
      if ( x[k] > max_x )
         max_x = x[k];
   }
   return max_x;
}
```

Read all instructions before beginning your work.

COMP1200-C - Assign 09
Due midnight – Thursday – April 16, 2015
Submit `assign09.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.txt` data file from the assign08 Announcement and save in your COMP1200/assign08 folder. If you do not have folders set up for your assignment files, this is a good time to start. assign09.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*: assign09.c**

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

Game Code	AU r-h-e*	Opp r-h-e*	no. innings	Attend	Game length hr : min
x	15-13-1	5-5-1	5	845	2:15
@	11-13-1	2-5-0	5	967	1:44

*r-h-e = runs-hits-errors

In assign09.c, you will re-write assign08.c using a 2D array for the save integer values and a double 1D array for the game time. Using the `fscanf` function, you will store ONLY the following information into 1D and 2D arrays:

- game code into a 1D `char` array;
- Auburn runs scored, Auburn runs allowed (Opp runs), number of innings, and attendance into a 2D `int` array;
- use hour and minutes to compute the game length and save into a 1D `double` array.

All other values are "skipped" using "hold" variables. See the `skipData.c` example on Canvas.
The printed summary should contain the number of games, minimum, mean, and maximum runs scored and allowed for all games and for SEC games. The game code for SEC games is an asterisk, *.

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 code[], int gameStats[][NUMCOLS], double hours[] );
```

Function opens and read the data file. Because code, a char, is the first for each game, put a blank at the beginning of the formatting, i.e. `" %c %d%c…"` If there is an open error, print a message and return to main. Read all data and save specified data in 1D and 2D 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 code[], int gameStats[][NUMCOLS], double hours[], int numGames );
```

Function uses the data in the arrays and the number of games to print a summary report. Do not use a variable name for the minimum, mean, and maximum value of an array. Instead "nest" (or use) the corresponding function as the `printf` argument.

```
int getSecGames ( char code[], int arrayAll[][NUMCOLS], int numGames, int arraySec[][NUMCOLS] );
```

Function uses the code array to find the SEC games. For each SEC game, store the game information in another 2D array. Return the number of SEC games. This function is used only used once because the 2D array has Auburn and opponents runs in SEC games.

```
int extraInnings( int gameStats[][NUMCOLS], int numGames, int extra[] );
```

Function searches the inning column for values greater than the number of innings. The game number of each game with extra innings is stored in another 1D array. Return the number of games with extra innings.

Create data analysis functions to return the minimum, mean, and maximum value of a column of a 2D integer array. Data analysis functions are given in the lecture slides. Note the addition of the column number parameter.

```
int    intColMax ( int x[][NUMCOLS], int n, int colNum );
int    intColMin ( int x[][NUMCOLS], int n, int colNum );
double intColMean( int x[][NUMCOLS], int n, int colNum );
```

Also, create a function to find and return the maximum value in a 1D double array.

***Problem Constants*:**

```
FILENAME    "2015_AU_softball.txt"
MAXGAMES    50  // estimated number of games in season
NUMSEC      25  // estimated number of SEC games in season
SECDODE     '*' // code for SEC games
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
```

New commands:
2D integer arrays
2D indexing
constant names used for all numbers

Revisit:
1D arrays

***Problem Inputs*:**
See above.

***Problem Outputs*:**
See above.

***Other variables*:**
As needed.

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 constant names NOT numbers throughout your program**.

-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 Output:

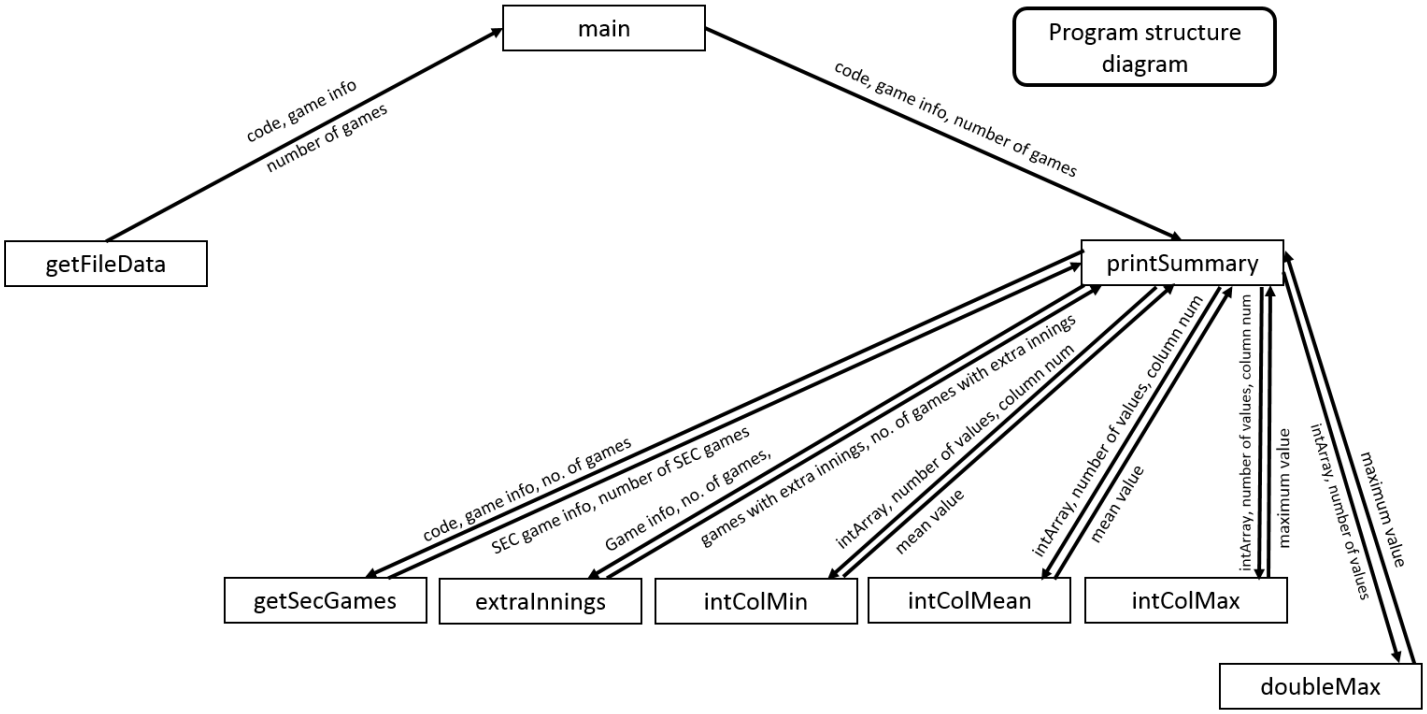
```
      2015 AU Softall Summary
                 #games Min Mean Max
Runs scored-all    41     2   8.9 20
Runs allowed-all          0   3.0 10
Runs scored-SEC    11     4   8.1 14
Runs allowed-SEC          0   4.1  8

Games with extra innings:
 23 38

Longest game played: 3.25 hours
```

***Submit via* Canvas:**

`assign09.c`	C program file


Program structure diagram
main
getFileData
printSummary
getSecGames
extraInnings
intColMin
intColMean
intColMax
doubleMax
code, game info
number of games
code, game info, number of games
code, game info, no. of games
SEC game info, number of SEC games
Game info, no. of games,
games with extra innings, no. of games with extra innings
intArray, number of values, column num
mean value
intArray, number of values, column num
mean value
intArray, number of values, column num
maximum value
intArray, number of values
maximum value