```c
// J Hundley
// assign07
// April 1, 2012
/*
- Store file name in a CONSTANT variable.
    Use the variable name in the program.
- Do not continue program if there is a problem opening file.
- Text file columns:
  1 - month, 2 - day, 3 - Opponent score, 4 - Auburn score, 5 - attendance
- Print a report of the season game results
    print * next to the highest attendance
    with legend for flags
- Print largest win/loss point spread with date and opponent
*/
#include <stdio.h>
#define FILENAME  "data07.txt"
#define MAXGAMES  40

// *****PROTOTYPE*****
   int maxInt(int array[], int count);
   int maxDiffIndex(int array1[], int array2[], int count);

   int main()
   {
      int numGames, g=0;    // games counters
      int month[MAXGAMES], day[MAXGAMES],       // date
          auScore[MAXGAMES], opScore[MAXGAMES], // scores
          attend[MAXGAMES], maxAttend;          // attendance
      int winSpreadIndex, lossSpreadIndex;      // index for max point spread

      FILE * filePtr;

      filePtr = fopen(FILENAME, "r");
     if (filePtr == NULL)  printf("File does not exist");

      else  // file does not exist
      {                          // good open continue with program
      // *****INPUT*****
      // read 2012 softball game results
         while(fscanf(filePtr, "%d %d %d %d %d", &month[g],&day[g],
                      &opScore[g],&auScore[g],&attend[g])!= EOF) g++;
         numGames = g;

      // *****COMPUTE*****
      // find max Attendence
         maxAttend = maxInt(attend, numGames);
      // find largest win and loss point spread
         winSpreadIndex  = maxDiffIndex(auScore, opScore, numGames);
         lossSpreadIndex = maxDiffIndex(opScore, auScore, numGames);

      // *****OUTPUT*****
      // print report of game results
      //    print headers
         printf("\n    2012 AUBURN TIGERS SOFTBALL\n");
         printf(" Auburn Games Results (as of %02d/%02d)\n\n",
                 month[numGames-1],day[numGames-1]);
         printf("    Date  Score W-L Attend\n");
         printf("    ----- ----- --- ------\n");
```

```
      //    print results for each game
         for (g=0; g<numGames; g++)
         {
         // print date, AU-Opp score
            printf("    %02d/%02d %02d-%02d",
                                       month[g],day[g],opScore[g],auScore[g]);

         // print W or L for win or loss
            if (auScore[g] > opScore[g])
               printf("  W  ");
            else
               printf("  L  ");

         // print attendance
            printf("%5d",attend[g]);
         // print flag for largest attendance
            if (attend[g] == maxAttend)
               printf("#\n");
            else
               printf(" \n");

         } // end for each game loop

      // print legend
         printf(" # largest game attendance\n\n");

      // print the date for largest win and loss point spread
         printf(" Largest point spread:\n");
         printf(" Win:  %2d on %02d/%02d\n",
            auScore[winSpreadIndex]-opScore[winSpreadIndex],
            month[winSpreadIndex],day[winSpreadIndex]);
         printf(" Loss: %2d on %02d/%02d\n",
            opScore[lossSpreadIndex]-auScore[lossSpreadIndex],
            month[lossSpreadIndex],day[lossSpreadIndex]);


      } //end else good file open

      return 0;
   }

// ========================================================
//   FUNCTIONS
// ========================================================

// find the max of a integer array
   int    maxInt(int array[], int count)
   {
      int c;
     int max_x;

      max_x = array[0];
      for (c=1; c<count; c++)
      {
         if (array[c] > max_x)
            max_x = array[c];
      }
      return max_x;
   }
```

```
// find the index of the largest difference
   int    maxDiffIndex(int array1[], int array2[], int count)
   {
      int c;
      int maxDiff, maxIndex;

      maxDiff = array1[0] - array2[0];
      for (c=1; c<count; c++)
      {
         if ( array1[c]-array2[c] > maxDiff)
         {
            maxDiff = array1[c]-array2[c];
            maxIndex = c;
         }
      }
      return maxIndex;
   }
```

COMP1200-CProg - assign 07
Due midnight – Wednesday – April 11
Submit `data07.txt` and `assignb07.c` via Blackboard

Before you start writing your program:

Read all of these instructions carefully. Submitting a development plan is not required for this assignment. I suggest that you create one and use it when writing your program.

***Problem*:**

***Program*: assign07.c**

Print a report of the Auburn 2012 softball season game results. The result statistics are saved in `data07.txt`.

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

Instructions:

- ☐ Insert comments at the top and throughout each file
  - Include the follow comments at the beginning of this (and ALL) files.
    // your name
    // assignment number
    // date you completed the assignment
    // statement(s) about collaboration
    // a short narrative about what the file does
  - Use your development plan as a guide for comments throughout each file
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Indent blocks.

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

- ☐ CONSTANT variables
  - Define a constant variable with the name of the file; use the variable name as the argument with the **`fopen()`**.
  - Define a constant variable with the maximum number of games. Use a number large enough for a full season including possible post season games, SEC playoff and a bowl game, but not too big. The program should work for any Auburn softball season.

- ☐ Using the data file
  - Protect your program from crashing by making sure that the file opens. If the file doesn't open properly, print an error message and end the program.

- ☐ Input
  - Read the Auburn 2011 season game results from `data07.txt`
  - There are five (5) columns of data in the data file.
    1 - month, 2 - day, 3 - opp score, 4 - AU score, 5 - attendance
  - Read the data into five (5) integer 1-D arrays. Note the arrays are parallel arrays, i.e., the result information of the first game is in the first element of each of the six arrays; the second game information in the second element of each array, etc.
  - Your program should work for any number of games in the file. Count the number of games.

- ☐ Computation
  - Use a user-defined function to find the largest game attendance.
  - Use a user-defined function to find the index of the game in which it the largest win and loss point spreads occurred. (Point spread is the difference between the Auburn and opponent score.)
  - Use the **max** user-defined function in the slides. The following user-defined function returns the index of the largest difference of the arrays. You will need to use it twice.

New commands/terms
1-D arrays
parallel arrays
read data into 1-D arrays
user-defined functions with 1-D array parameters
print with leading zeros

Revisit
CONSTANT variable

```
// find the index of the largest difference of two integer arrays
   int maxDiffIndex(int array1[], int array2[], int count)
   {
      int c;
      int maxDiff, maxIndex;

      maxDiff = array1[0] - array2[0];
      for (c=1; c<count; c++)
      {
         if ( array1[c]-array2[c] > maxDiff)
         {
            maxDiff = array1[c]-array2[c];
            maxIndex = c;
         }
      }
      return maxIndex;
   }
```

- ☐ Output
  - Refer to the Sample Output for the information that should be included in the report.
    - Include the titles and column headings. NOTE: the "as of" date should be obtained from the last month and day in the arrays.
    - MULTIPLE `printf()`s CAN BE USED TO PRINT ONE LINE.
    - Depending on whether Auburn wins or loses, print "W" or "L".
    - Print a flag "#" next to the attendance equal the largest attendance else print a space.
  - Print a legend that indentifies the flags in the report.
  - Print the largest win and loss point spread with the date of the game.
- ☐ Printing
  - Column numbers **right-justified**, i.e., right-aligned
  - Print the month, day, and scores with leading zeros.

Sample Input/Output: ***<u>THESE ARE PARTIAL LISTS.</u>***

```
   2012 AUBURN TIGERS SOFTBALL
Auburn Games Results (as of 03/25)
     Date  Score W-L Attend
     ----- ----- --- ------
     02/09 00-06  W    252
     02/10 04-12  W    155
     02/10 02-10  W    185
     02/11 00-02  W    176
     02/12 02-06  W    104
     02/17 00-08  W    508
     02/19 00-01  W    345
     02/19 05-12  W    415
     . . .
     03/11 00-08  W    213
     03/14 00-02  W   1226#
     03/16 00-02  W    813
     03/17 06-03  L   1015
     03/18 04-02  L   1217
     03/21 07-01  L    438
     03/23 08-04  L    291
     03/24 05-02  L    540
     03/25 01-02  W    364
# largest game attendance
Largest point spread:
Win:  10 on 03/03
Loss:  6 on 03/21
```

data07.txt

```
2    9    0    6    252
2    10   4    12   155
2    10   2    10   185
2    11   0    2    176
2    12   2    6    104
2    17   0    8    508
2    19   0    1    345
2    19   5    12   415
. . .
3    3    1    0    154
3    3    1    11   215
3    4    1    2    217
3    11   0    8    213
3    14   0    2    1226
3    16   0    2    813
3    17   6    3    1015
3    18   4    2    1217
3    21   7    1    438
3    23   8    4    291
3    24   5    2    540
3    25   1    2    364
```

Submit via Blackboard:

`assign07.c`	C program file
`data07.txt`	Data file