PROBLEM: This program uses structures and pointers.
It will receive (via a file) a set of long jump tries for a set of jumpers. To cut down on the complexity of this program, we will use a zero for both fouled jumps and passed jumps. We will figure out the best jump for each competitor, the average of the best jumps, and the winning best jump.

(**1**) You first need to declare a structure type *jumper_t*.
I named my structure *jumper_t* and its 4 parts are:
a character array *name* that is 16 in length,
a double array of *tries* that is N_TRIES in length,
a double named *best_jump*, and
a double named *deviation*.

name	**try 0**	**try 1**	**try 2**	**try 3**	**try 4**	**try 5**	best_jump	deviation

Array *tries*

(**2**) Next you need to declare a structure type *stats_t*.
I named my structure *stats_t* and its 2 parts are:
variables, both type double, named *average_of_best*, *winning_jump* .

(**3**) Write the function *get_stats*. The prototype is:

```
void get_stats(jumper_t jump_list[NCOMPETITORS], /* in & out */
               stats_t *jump_stats);             /* output   */
```

It will now figure the best jump for each jumper, compute the all-over average of all the best jumps, find the winning jump, and each jumper's deviation from the winning jump.

(**4**) You will be provided a test driver program that needs very little changing. You will only need to add the two structures and the one function *get_stats*.

FILES TO COPY:
First move to your class folder by typing: **cd csc60**
The following command will create a directory named **lab7** and put all the needed files into it below your csc60 directory.
Type: **cp -R /gaia/home/faculty/bielr/files_csc60/lab7 .**
Spaces needed: (1) After the **cp** ↑ *Don't miss the space & dot.*
(2) After the **-R**
(3) After the directory name at the end & before the dot.
After the files are in your account and you are still in **csc60**, you need to type: **chmod 755 lab7**
This will give permissions to the directory.
Next move into lab7 directory, and type: **chmod 644 lab7***
This will give permissions to the files.
Your new lab7 directory should now contain: lab7.c, lab7.dat, lab7sample.dat

INPUT/OUTPUT DESCRIPTION:

The program input is a set of jumper's names and their six tries the long jump. The jump lengths are type double.

Each record/line of the file has a jumper’s name and six tries, and space for the best_jump of the jumper and the deviation from the winning jump.

The file consists of the four names, followed by the first six tries, and then each successive set of tries.

ALGORITHM DEVELOPMENT - Pseudocode:

```
/*------------------------------------------------------------*/
main
   out_file = open_out_file ();
   get_data(IN_FILENAME, jump_list);
   get_stats(jump_list, &jump_stats);
   print_all(out_file, jump_list, &jump_stats);

/*------------------------------------------------------------*/
FILE * open_out_file(void)
   Open the output file
   Return the output file pointer.

/*------------------------------------------------------------*/
void get_data (char *filename,                /* input  */
           jumper_t jump_list[NCOMPETITORS] );  /* output */
  Open the appropriate file
  Read the data into the jump_list array
         (Use double loops, one each for columns and rows)
  Close the file
/*------------------------------------------------------------*/
```

➔ MORE ON NEXT PAGE

```
/*---------------------------------------------------------------*/
/*    This is a sub-function that you have to write */
void get_stats( jumper_t jump_list[NCOMPETITORS],     /* in & out */
                     stats_t *jump_stats )                          /* in & out */
{
     Zero out the average_of_best.  (HINT: use the -> notation)
     Zero out the winning_jump.

     loop from r=zero to r< NCOMPETITORS, increment by one  {
          set the jumper's best_jump to the jumper's first jump
          loop from c=one to c< N_TRIES increment by one
               if the next jump TRY  >  the contents of the best jump column
                    set the best column to this better jump try
          }          /* end of the loop using "c" */
          add the jumper's best jump into the running total average_of_best

         loop from c=zero to c< N_TRIES increment by one {
              figure the winning-best jump of all the jumps (use an IF)
         }           /* end of the second loop using "c"  */
     }               /* end of the loop using "r"  */
    compute the average of the best jumps

    loop from r=zero to < NCOMPETITORS increment by one {
         figure the jumper's deviation from the winning_jump
         (deviation is all-over winning_jump minus each jumper's best jump)
    }                /* end of the second loop using "r"  */
    return
}
/*---------------------------------------------------------------*/
/*    The print_all SUB-FUNCTION is provided for you          */
void print_all(FILE * out_file,
                    jumper_t jump_list[NCOMPETITORS] ,
                    stats_t *jump_stats )
```

/* This routine does all the printing and ***is a model for referencing the variables***. */

```
/*---------------------------------------------------------------*/
```

➔ more on next page

REMINDER: Check the validity of your answers.

If a parameter is passed into a function without the asterisk, you are to use the "dot" notation.

If the parameter is passed into a function with the asterisk, you are to use the "points into" notation (->).

HAND EXAMPLE:
This is a sample hand example. It may not match the provided data file in length or in value! This sample data is in the file named prog5sample.dat.

SAMPLE OUTPUT:

```
Your Name, Lab 7 output.

Track Results

Name               Try 1   Try 2   Try 3   Try 4   Try 5   Try 6   Best Jump   Deviation
----------------   -----   -----   -----   -----   -----   -----   ---------   ---------
Moe Johnson         1.00    5.00    2.00    3.00    0.00    3.00      5.00        1.00
Lad Loop            2.00    6.00    2.00    4.00    6.00    2.00      6.00        0.00
Monroe Miner        3.00    2.00    2.00    5.00    5.00    0.00      5.00        1.00
Jay Jacob           4.00    2.00    0.00    3.00    4.00    1.00      4.00        2.00


Best Jump Average  =   5.00 meters

Winning Jump       =   6.00 meters
```

Comments:

In function get_data, I had a lot of trouble getting the columns to line up. When I used a conversion specifier of %15c, the names printed out, but each line of numbers moved one space to the left. So then I tried %s. The string had a New-Line buried in it, resulting in a name on one line, and its numbers on the next line.

So finally, I read the string using "fgets", function-get-string. The string still contains the New-Line, but I also used the function "strchr" to seach the string for the location of the New-Line, and changed that position in the string to a NULL. Everything then printed out OK.
(One can do a "man fgets" for more information.)

A string, by definition, is a character array where the last character is NULL.
That is why there is code in get_data to put a NULL into the last location of the string.

➔ more on next page

PREPARE YOUR FILE FOR GRADING:

Make sure your program has been set to use ***lab7.dat*** *and has been re-complied (gcc).*

When all is well and correct,

Type: **script StudentName_lab7.txt** [Script will keep a log of your session.]

Type: **cat lab7.c** to display the code in your session.

Type: **gcc lab7.c** to compile the code

Type: **a.out** to run the program (If you use a *-o name,* then use that name to execute.)

Type: **cat lab7.out** to show contents of the output file

Type: **exit** to leave the script session

Turn in your completed session:

Go to Canvas and turn in your script session (StudentName_lab7.txt).