```
% J Hundley
% assign10.m
% read Auburn season football scoring stats from spreadsheet
% create a summary report and spreadsheet of at home games
clc, clear all
format compact
%***** CONSTANTS *****
AU_STATS  = 'AU_stats10.xls';
IN_SHEET  = 'AU season';
OUT_SHEET = 'AU home games';
%***** INPUT *****
if ~exist( AU_STATS, 'file' )
    disp( 'file not available' )
else
    % file available continue with problem solution
    % read auburn statistics
    [ numbers, strings ] = xlsread( AU_STATS, IN_SHEET );
    %***** OUTPUT *****
    % print report and build array of at home games
    cellArray = createReportCellArray( numbers, strings );
    % write cell array to spreadsheet
    xlswrite( AU_STATS, cellArray, OUT_SHEET  )
end

========================================================== SAVED IN createReportCellArray.m
% J Hundley
% createReportCellArray.m
function outCell = createReportCellArray( stats, labels )
% print summary report and build cell array of at home games
printReport( stats, labels )
outCell = createCellArray( stats, labels );
end
================================================== SUBFUNCTION IN createReportCellArray.m
function [] = printReport( stats, labels )
% print report of at home games
[nGames,ncols] = size( stats );
% season record SEC and overall
fprintf( '%s Summary\n', labels{1,1} )

% overall season record
wins = find( stats(:,1) > stats(:,2) );                    % can count in loop
fprintf( 'Overall season record: %d-%d \n', length(wins), nGames-length(wins) )

% SEC season record
numSecGames = 0;
numSecWins  = 0;
for g = 1:nGames
    if ~isempty( strfind( labels{g+2,2}, '*' ) )
        numSecGames = numSecGames + 1;
        if stats(g,1) > stats(g,2)
            numSecWins = numSecWins + 1;
        end
    end
end % end for g=
fprintf( 'SEC season record: %d-%d \n', numSecWins, numSecGames-numSecWins )

% game with largest point spread
[maxSpread, gameNum] = max( abs( stats(:,1)-stats(:,2) ) );
fprintf( 'Largest point spread was %d on %s against %s.  \n',...
    maxSpread, labels{gameNum+2,1}, labels{gameNum+2,3}(4:end) )
end % end function printReport
```

```
================================================ SUBFUNCTION IN createReportCellArray.m
function outCell = createCellArray( stats, labels )
% build array of at home games
[nGames,ncols] = size( stats );
%***** COMPUTE *****
% compute time in hours
hours = stats(:,3) + stats(:,4)/60;
aveAttend = mean( stats(:,5) );

% build title and headers rows                    sprintf() or build a string using []
outCell{1,1} = sprintf( '%s Home Games as of %s \n', labels{1,1},labels{nGames+2} );
outCell{2,1} = labels{2,1};
outCell{2,2} = labels{2,3};
outCell{2,3} = labels{2,4};
outCell{2,4} = labels{2,5};
outCell{2,5} = 'W-L';
outCell{2,6} = 'Hours';
outCell{2,7} = labels{2,8};

% build stats rows for home games (can be by column)
cellRow = 3;
for g = 1:nGames
    if strfind( labels{g+2,3}, 'vs' )
        outCell{cellRow,1} = labels{g+2,1};
        outCell{cellRow,2} = labels{g+2,3};
        outCell{cellRow,3} = stats(g,1);
        outCell{cellRow,4} = stats(g,2);
        if stats(g,1) > stats(g,2)
            outCell{cellRow,5} = 'W';
        else
            outCell{cellRow,5} = 'L';
        end
        outCell{cellRow,6} = hours(g);
        outCell{cellRow,7} = stats(g,5);
        if stats(g,5) > aveAttend
            outCell{cellRow,8} = '*';
        end
        cellRow = cellRow + 1;
    end % end if stats
end % end for g=
outCell{cellRow,1} = 'Average attendance:';
outCell{cellRow,7} = aveAttend;
end % function createCellArray
```

Input for grading:

	A	B	C	D	E	F	G	H
1	2019 Auburn Football Season							
2	Date	SEC	Opponent	AU	Opp	Hrs	Min	Attend
3	8/31/2019	*	vs Oregon	27	21	3	38	60662
4	9/7/2019		vs Tulane	24	6	3	27	85317
5	9/21/2019	*	at Texas A&M	28	20	3	27	101681
6	10/5/2019	*	at Florida	13	35	3	31	90584
7	11/2/2019		vs Ole Miss	20	14	3	10	60000
8	11/16/2019	*	vs Georgia	14	21	3	42	87451

AU season | AU home gam ...

```
==============================================
2019 Auburn Football Season Summary
Overall season record: 4-2
SEC season record: 2-2
Largest point spread was 22 on 10/5/2019 against Florida.
```

Output for grading:

	A	B	C	D	E	F	G	H
1	2019 Auburn Football Season Home Games as of 11/16/2019							
2	Date	Opponent	AU	Opp	W-L	Hours	Attend	
3	8/31/2019	vs Oregon	27	21	W	3.633333	60662	
4	9/7/2019	vs Tulane	24	6	W	3.45	85317	*
5	11/2/2019	vs Ole Miss	20	14	W	3.166667	60000	
6	11/16/2019	vs Georgia	14	21	L	3.7	87451	*
7	Average attendance:						80949.167	

AU season | AU home games

Average attendance: may be in column 1 OR **Average** in column 1 and **attendance:** in column 2.

Read all instructions before beginning your work.

COMP1200-MatLab - assign 10
Due 4:45 pm – Friday – November 22, 2019
Submit `assign10.m` and `createReportCellArray.m` **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 the complete instructions.

Submit BEFORE leaving for Thanksgiving break

FINAL EXAM during DEC. 4 class

***Program*: assign10.m**
Scoring statistics are available for Auburn for each football game. Read the statistics from the input spreadsheet and print a report and write an output spreadsheet as instructed.

***Problem Constants*:**
AU file name `'AU_stats10.xls'` **<< NOT .xlsx**
Input spreadsheet `'AU season'`
Output spreadsheet `'AU home games'`

***Problem Inputs*:**
See input spreadsheet. **Note, you do NOT know the number of games.**
You may observe the matrix results of the `xlsread()` by temporarily removing the semicolon.

Problem Outputs:
Report and spreadsheet in input file. See output.

Other variables:
As needed.

Equations:
As needed.

Output:
See report and spreadsheet below.

	A	B	C	D	E	F	G	H
1	2019 Auburn Football Season							
2	Date	SEC	Opponent	AU	Opp	Hrs	Min	Attend
3	8/31/2019		vs Oregon	27	21	3	38	60662
4	9/7/2019		vs Tulane	24	6	3	27	85317
5	9/14/2019		vs Kent State	55	16	3	9	84542
6	9/21/2019	*	at Texas A&M	28	20	3	27	101681
7	9/28/2019	*	vs Mississippi State	56	23	3	30	87451
8	10/5/2019	*	at Florida	13	24	3	31	90584
9	10/19/2019	*	at Arkansas	51	10	3	16	54619
10	10/26/2019	*	at LSU	20	23	3	52	102160
11	11/2/2019	*	vs Ole Miss	20	14	3	10	87457

AU season

In `assign10.m`,
Read the input spreadsheet.
Use `createReportCellArray()`
to print report using `printReport()`
to create and return cell array using `createCellArray()`
Use `xlswrite()` to write the cell array returned from `createReportCellArray()`

MACs may not allow `xlswrite()`. There is the reason for the warning in the syllabus. You may create your output cell array; then go to a lab to test the `xlswrite()`.

Functions should be named as given
Variable names may be different, but the order and quantity should be as given.

Primary function:
```
function outCell = createReportCellArray( stats, labels )
```
Use the two subfunctions only.

If you get the following error, your .xls file is open. You cannot write to an open file. Close the file and rerun.
```
Error using xlswrite (line _)
The file D:\COMP 1200\ ...
\assign10\AU_stats10.xls is
not writable.  It might be
locked by another process.
```

Subfunction used by `createReportCellArray()`:
```
function [] = printReport( stats, labels )
```
Print a report as shown below that contains:
Title (use title from the spreadsheet) and labels.
Auburn's overall season record: #wins-#losses
Auburn's SEC season record: #SEC wins-#SEC losses
Largest point spread with date and opponent for the game with the largest point spread.
All point spreads should be a possible value.
Use opponent cell for opponent name. Note: each character in a string is a column.

Subfunction used by `createReportCellArray()`:
```
function outCell = createCellArray( stats, labels )
```
As available, use cells from input spreadsheet in the output spreadsheet for title and headers; otherwise use a string literal.
ONLY include home games in the output spreadsheet. Use input information for date, opponent, scores, and attendance.
Compute output hours using input hours and minutes.
Mark the above attendance with an *.
In the row after last game add a label and average attendance.

Report:

```
2019 Auburn Football Season Summary
Overall season record: 7-2
SEC season record: 4-3
Largest point spread was 41 on 10/19/2019 against Arkansas.
Warning: Added specified worksheet.
> In xlswrite>activate_sheet (line 298)
  In xlswrite/ExecuteWrite (line 264)
  In xlswrite (line 218)
  In assign10 (line 24)
```

Message seen with first successfully write to spreadsheet.

Do not use commands and statements beyond what has been taught on class.

New commands:
Continue ONLY if file exist.
`xlsread()` from input spreadsheet
`xlswrite()` with output spreadsheet
cell indexing `{}`
`strfind(), abs(), end`

Use data to determine number of games.
HINT: the game number and output cell row counter as not the same number.

Continue to use:
Functions and commands as needed
Use descriptive variables.
Remove statements not needed for

Spreadsheet as first viewed

	A	B	C	D	E	F	G	H
1	2019 Auburn Football Season Home Games as of 11/2/2019							
2	Date	Opponent	AU	Opp	W-L	Hours	Attend	
3	########	vs Oregon	27	21	W	3.633333	60662	
4	9/7/2019	vs Tulane	24	6	W	3.45	85317	*
5	########	vs Kent St	55	16	W	3.15	84542	*
6	########	vs Mississ	56	23	W	3.5	87451	*
7	########	vs Ole Mis	20	14	W	3.166667	87457	*
8	Average attendance:						83830.33	

AU season | AU home games

Date of last game in spreadsheet You do not know the number of games.

	A	B	C	D	E	F	G	H
1	2019 Auburn Football Season Home Games as of 11/2/2019							
2	Date	Opponent	AU	Opp	W-L	Hours	Attend	
3	8/31/2019	vs Oregon	27	21	W	3.633333	60662	
4	9/7/2019	vs Tulane	24	6	W	3.45	85317	*
5	9/14/2019	vs Kent St	55	16	W	3.15	84542	*
6	9/28/2019	vs Mississ	56	23	W	3.5	87451	*
7	11/2/2019	vs Ole Mis	20	14	W	3.166667	87457	*
8	Average attendance:						83830.33	

AU season | AU home games

Spreadsheet after adjusting columns.

Instructions for all assignment scripts: ALL script files

- ☐ See Standards for Documentation of MATLAB Programs on the Canvas Resources page.
- ☐ Insert comments at the top and throughout each file.
  - o Include the follow comments at the beginning of this (and ALL) files.
    % submitter's name, **GROUP # or "none"**
    % other group members' names or "none"
    % **`program file name`**, ex. assign02a.m
    % due date of the assignment
    % **statement about collaboration REQUIRED.**
    % a short narrative about what the file does
  - o Use the algorithm given as comments throughout your program.
- ☐ Observe the instructor's rule for naming variables.
  - o Use ALL CAPS for constants variable names.
  - o Start other variables with lower case.
  - o Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Code clarity:
  - o Indent blocks as needed. **Use Smart Indent.**
  - o Divide your solution program code into sections as noted in the algorithm. Use blank lines as needed to group statements.
  - o Use section comments as well as the algorithm step comments.
  - o Remove statements from previous assignments that do not apply to the current requirements.
- ☐ Use comments to show units.
- ☐ **Use the CONSTANT and variable names, not numbers. Exceptions are incrementers (or counters) and numbers without identity.**
- ☐ No extra output, i.e. use semicolons

GRADE OF ZERO **for a file if submitter name not part of Canvas group.**

(-3pts) No CURRENT GROUP# or "none".
(-3pts) For your own protection, type "none" for other group members if submitting alone.
(-5pts) Five point penalty for not joining your Canvas group.

(-5pts) Zero points for comments if no collaboration statement.

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

Submit via Canvas:

`assign10.m`	MATLAB script file
`createReportCellArray.m`	MATLAB function file