```
% J Hundley
% assign06a.m
%{
   You can use trigonometry to find the height of a building.
   Suppose you measure the angle between the line of sight and the
   horizontal line connecting the measuring point and the building.
   You can calculate the height of the building with the following formulas:
        tan(theta) = height / distance
        height = distance * tan(theta)
   Assume that the distance to the building along the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees
   plus/minus 3 degrees.
   --Read degrees using fscanf() from data file into a vector degrees.
a) compute the height of the building for given the distance and each acceptable angle
   display a label and the table of angles and heights from the results of a).
b) get and print average angle in the vector
%}

clc, clear all
%*****CONSTANT*****
DISTANCE     = 120;                              % meters
NUM_DEGREES  = 10;
DEGREES_FILE = 'dataAssign06.txt';

%*****INPUT*****
[ fileID, msg ] = fopen( DEGREES_FILE, 'r' );
if fileID < 0
    fprintf( 'File open error\n' )
else
    for d = 1: NUM_DEGREES
        theta(d) = fscanf(fileID, '%f', 1);     % degrees
    end

    %*****COMPUTATION*****
    % compute the height of the building for given distance and each angle
    heightAll = DISTANCE .* tand( theta );      % meters

    % get average angle of angles in vector
    meanTheta  = mean( theta );                 % degrees
    % compute the height of the building using the average angle
    heightMean = DISTANCE .* tand( meanTheta ); % meters

    %*****OUTPUT*****
    % print title, headers and table of angles and heights
    fprintf(' Angle     Height\n')                          One or more fprintf() ok for whole table.
    fprintf('Degrees    Meters\n')
    fprintf(' %5.2f     %5.2f \n', [theta; heightAll] )     NO loop fprintf() for number columns
                                                              Variable or [] for column argument ok
    % print a average angle and height using average angle
    fprintf('\nThe height %5.2f meters for the average angle %5.2f degrees.\n',...
        heightMean, meanTheta )

end % end good file open
```

```
% J Hundley                                                    assign06b
% assign06b.m
%{
   You can use trigonometry to find the height of a building.
   Suppose you measure the angle between the line of sight and the
   horizontal line connecting the measuring point and the building.
   You can calculate the height of the building with the following formulas:
        tan(theta) = height / distance
        height = distance * tan(theta)
   Assume that the distance to the building along the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees
   plus/minus 3 degrees.
   --Read degrees using load() from data file into a vector degrees.
a) compute the height of the building for given the distance and each acceptable angle
   display a label and the table of angles and heights from the results of a).
b) get and print average angle in the vector
%}

clc, clear all
%*****CONSTANT*****
DISTANCE     = 120;                              % meters
DEGREES_FILE = 'dataAssign06.txt';

%*****INPUT*****
if ~exist( DEGREES_FILE, 'file' )
    fprintf( 'File not found\n' )
else
    % read data file
    theta = load( DEGREES_FILE );                % degrees

    %*****COMPUTATION*****
    % compute the height of the building for given distance and each angle
    heightAll = DISTANCE .* tand( theta );       % meters

    % get average angle of angles in vector
    meanTheta  = mean( theta );                  % degrees
    % compute the height of the building using the average angle
    heightMean = DISTANCE .* tand( meanTheta ); % meters

    %*****OUTPUT*****
    % print title, headers and table of angles and heights
    fprintf( ' Angle     Height\n' )                          One or more fprintf() ok for whole table.
    fprintf( 'Degrees    Meters\n' )
    fprintf( ' %5.2f     %5.2f \n', [theta; heightAll] )      NO loop fprintf() for number columns
                                                                Variable or [] for column argument ok
      % print a average angle and height using average angle
    fprintf('\nThe height %5.2f meters for the average angle %5.2f degrees.\n',...
        heightMean, meanTheta )
      % print the number of degrees read
    fprintf( '%.0f degrees were read from the data file.\n', length( theta ) )

end % end good file open
```

Read all instructions before beginning your work.

COMP1200-MatLab - assign 06
Due 4:45 pm – Friday – October 18, 2019
Submit `assign06a.m` and `assign06b.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. You may use statements from your assign05 but remove all statements that do not apply to assign06.

***Problem*:**

Trigonometry can be used to find the height of a building. Suppose you measure the angle between the line of sight and the horizontal line connecting the measuring point and the building. You can calculate the height of the building with the following formulas:

`tan(theta)=h/d` `h=d*tan(theta)`

Assume that the distance to the building along the ground is 120 meters and the angle measured along the line of sight is 30 degrees plus/minus 3 degrees.

***Program*: assign06a.m**

Using the requirements below, **create an algorithm**. **Type your algorithm as comments** in your script and use them as a guide as you type statements to accomplish the requirements.
START YOUR SCRIPT WITH THE REQUIRED COMMENTS GIVEN IN ASSIGNMENTS 2-4. See below.

***** CONSTANTS *****

***** INPUT *****
If the file is available (i.e. `fopen()`) correctly, continue with the following instructions,
otherwise `fprintf()` a message ONLY.
In a `for` loop with the CONSTANT number of degrees to form the control vector,
use `fscanf()` to read the angles from the data file into a vector one number at a time.

***** COMPUTE *****
Compute the heights of the building for given the distance and each angle.
Compute the height of the building using the distance and the **average angle**.

***** OUTPUT *****
`fprintf()`a table of angle and heights with column headers.
`fprintf()`the **average angle** and the height computer using the **average angle**.

***Problem Constants*:**

```
DISTANCE        120 % meters
NUM_DEGREES     10
DEGREES_FILE    'dataAssign06.txt'
```

***Problem Inputs*:**
angles (theta) in degrees from data file

Problem Outputs:
heights of the building for given the distance and angles
average angle
height of the building using the **average angle**

Other variables:
as needed

Equations:
See above.

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

New commands:
`fopen(), fscanf()`
`fprintf()`

Continue to use:
Do not use `disp()`
Use descriptive variables.

Output:

```
Angle      Height
Degrees     Meters
 28.72      65.75
 31.54      73.65
 31.52      73.59
 29.28      67.29
 30.41      70.43
 27.46      62.36
 27.32      61.99
 30.18      69.79
 31.68      74.06
 32.60      76.74

The height 69.48 meters for the average angle 30.07 degrees.
```

When using `fprintf()`, you have control over the spacing of columns by using blank spaces.
Pay close attention to how the numbers in the vectors are printed.
Horizontal and vertical vectors are handled differently. See examples in text.
*** **ONE `fprintf()` NOT in a loop for columns of numbers**

***Program*: assign06b.m**

To get started, save your assign06a.m as assign6b.m.
Remove and add statements as needed for assign6b.m. Note: the constant `NUM_DEGRESS` is not longer needed.
Edit comments as needed.

***** INPUT *****
If the file is `exist()`, continue with the following instructions,
otherwise `fprintf()` a message ONLY.
Use `load()` to read the angles from the data file into a vector. All values will be assigned to the vector at once.

***** OUTPUT *****
Same as assign006a plus:
`fprintf()`the number of angles read in a statement by imbedding a **`length()`** function for the data to print.

Output:

```
 Angle     Height
Degrees    Meters
 28.72     65.75
 31.54     73.65
 31.52     73.59
 29.28     67.29
 30.41     70.43
 27.46     62.36
 27.32     61.99
 30.18     69.79
 31.68     74.06
 32.60     76.74

The height 69.48 meters for the average angle 30.07 degrees.
10 degrees were read from the data file.
```

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

New commands:
`exist(), load()`
`length()`

Continue to use:
`fprintf()`
Do not use `disp()`
Use descriptive variables.

Instructions for all assignment scripts:

- ☐ See Standards for Documentation of MATLAB Programs on the Canvas Resources page.
- ☐ 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 # 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
  - Use the algorithm given as comments throughout your program.
- ☐ Observe the instructor's rule for naming variables.
  - Use ALL CAPS for constants variable names.
  - Start other variables with lower case.
  - Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Code clarity:
  - Indent blocks as needed. **Use Smart Indent.**
  - Divide your solution program code into sections as noted in the algorithm. Use blank lines as needed to group statements.
  - Use section comments as well as the algorithm step comments.
  - 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:

`assign06a.m`	MATLAB script file
`assign06b.m`	MATLAB script file