```
% J Hundley
% assign02a.m
% find the gravitation force from sun to Earth, Venus and Sun

clc, clear all
format long
format loose

% ***** CONSTANTS *****
% get the universal gravitational constant
GRAVITY = 6.67384e-11;       % Newtons

% ***** INPUT *****
% get the mass of Earth, Venus and Sun
massEarth = 5.97e24;   % kg
massVenus = 4.87e24;   % kg
massSun   = 1.9891e30; % kg

% get the distant from Earth to Sun
distEarthSun = 149.6e9; % m
% get the distant from Venus to Sun
distVenusSun = 108.2e9;  % m

% ***** COMPUTE *****
% compute force for Earth from the Sun
forceEarth = (GRAVITY * massEarth * massSun) / distEarthSun^2; % Newtons
% compute force for Venus from the Sun
forceVenus = (GRAVITY * massVenus * massSun) / distVenusSun^2; % Newtons

% ***** OUTPUT *****
% display the Gravitation force from Sun to Earth
disp( 'Gravitation force from Sun to Earth (Newtons):' )
disp( forceEarth )
% display the Gravitation force from Sun to Venus
disp( 'Gravitation force from sun to Venus (Newtons):' )
disp( forceVenus )
```

```
% J Hundley
% assign02b.m
% find the gravitation force from sun to planets

clc, clear all
format long
format compact

% ***** CONSTANTS *****
GRAVITY = 6.67384e-11;       % Newtons

% ***** INPUT *****
% get mass and distance vectors for Mercury, Venus, Earth
planetMass     = [ 3.30e23 4.87e24 5.97e24 ];  % kg
planetSunDist  = [ 57.9e9 108.2e9 149.6e9 ];   % km

% get mass for sun
massSun        = 1.989e30;                     % kg

% ***** COMPUTE *****
% compute vector forces from sun to all planets
force = GRAVITY .* planetMass .* massSun ./ planetSunDist .^2; % Newtons
% create table: mass, distant, force
table = [ planetMass', planetSunDist', force' ];

% ***** OUTPUT *****
% display table with title and column headers
disp( '   Gravitation force from sun to planets (Newtons):' )
disp( '   Mass (kg)           Distant (m)         Force (N)' )
disp( table )
```

*Read **all** instructions before beginning your work.*

COMP1200-MATLAB - assign 02
Due 4:45pm – Friday – September 13, 2019
Submit `assign02a.m` and `assign02b.m` **via Canvas**

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.

You will be instructed to solve a problem in two parts and submit files for each part. This demonstrates how to approach solving a large problem by solving one smaller part at a time. By solving a smaller part correctly before adding the next, one can keep the number of statements and errors that may result from them to a minimum. This approach also demonstrates how an existing problem may change in scope and thus the solution program must be modified. By saving the first part with an incremental file name, additional versions can easily be saved using subsequent names providing a good backup file.

***Before you start writing your program**:*
Read the complete instructions including the **algorithm**. An **algorithm** contains the steps needed to guide you through solving a problem. Use the **algorithm** as comments to a guide you when writing the MATLAB program file solution for the following problem.

***Problem*:**
The Force of Gravity equation computes the gravitational force between the two masses: the Sun and the planet or other body from our solar system. In this equation, the masses are treated as point masses separated by a specified distance. The Universal Gravitational Constant is 6.67384E-11 N.

where:

$$F = G\frac{m_1 m_2}{r^2}$$

- F is the force between the masses (Newtons),
- G is the gravitational constant (N),
- m_1 is the first mass (kilograms),
- m_2 is the second mass (kilograms), and
- r is the distance between the centers of the masses (meters).

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.

***Program*: assign02a.m**
You are to write a program (a MATLAB script file) that uses the masses of planets, mass of Sun, and distances from Sun to planets to compute and display the force of the Sun to each planet with appropriate labeling.

***Problem* CONSTANTS:** (with units)

GRAVITY	6.67384e-11	% Newtons

***Problem Inputs*:** (with units)

Mass of Earth	5.97e24	% kilograms
Mass of Venus	4.87e24	% kilograms
Mass of Sun	1.9891e30	% kilograms
Distance from Earth to Sun	149.6e9	% meters
Distance from Venus to Sun	108.2e9	% meters

Problem Outputs: (with units)
Force in Newtons
Other variables: (with units)
none
Equation:
See above.
Algorithm: See green comments below.
Type the comments in the editor window and use as a guide for typing the MATLAB statements.

NO error checking.
Do not use commands and statements beyond what has been taught on class.

New commands:
`format long, loose`
`disp()`
Use descriptive variables.

Run output:

```
Gravitation force from Sun to Earth (Newtons):
     3.541143709860876e+22

Gravitation force from sun to Venus (Newtons):
     5.522132812796184e+22
```

Some rounding differences allowed.

Start your program file by typing the following into your empty editor window.

- Type yours/your group and other required information comments.
- Type the algorithm as given below as comments to guide you when writing the MATLAB instructions to do the tasks to solve the given problem.
- Below the comment, type the MATLAB statement(s) that do what the comment says. This example should help. →

A sample program
The algorithm was added as comments to guide you when writing the MATLAB program file solution for the problem.

A letter is not a descriptive name, but in this equation a, b, c are commonly used as quad equation coefficients.

```
% submitter's name, GROUP # or none
% other group members' names or none
% program file name
% due date of the assignment
% statement(s) about collaboration. See syllabus for examples.
% a short narrative about what the file does

clc, clear all
format long
format loose

% ***** CONSTANTS *****
% get the universal gravitational constant

% ***** INPUT *****
% get the masses of Earth, Venus and Sun
% get the distance from Earth to Sun
% get the distance from Venus to Sun

% ***** COMPUTE *****
% compute force from Sun to Earth
% compute force from Sun to Venus

% ***** OUTPUT *****
% display the Gravitation force from Sun to Earth
% display the Gravitation force from Sun to Venus
```

```
clc, clear all
format short

%*****INPUT*****
% Get coefficients
a = 1;   % 1st coefficient
b = 3;   % 2nd coefficient
c = -10; % 3rd coefficient

%*****COMPUTE*****
% Calculate the roots
root1 = (-b + sqrt(b^TWO - FOUR*a*c)) /(TWO*a);
root2 = (-b - sqrt(b^TWO - FOUR*a*c)) /(TWO*a);

%*****OUTPUT*****
% Display coefficients and roots
disp('Given coefficients:')
disp(a)
disp(b)
disp(c)
disp('The real roots are:')
disp(root1)
disp(root2)
```

***Program*: assign02b.m**

You are to write a program (a MATLAB script file) that uses the masses of planets, mass of Sun, and distances from Sun to planets to compute and display the force of the Sun to each planet with appropriate labeling.

	MERCURY	VENUS	EARTH
Mass (10^{24}kg)	0.330	4.87	5.97
Distance from Sun (10^{6} km)	57.9	108.2	149.6
Force of sun's gravity (10^{22} N)	1.30637878	5.53822703	3.54146321

***Problem* CONSTANTS:** (with units)

```
GRAVITY                          % Newtons
```

***Problem Inputs*:** (with units)

```
Vector of planet masses          % kilograms
Mass of Sun                      % kilograms
Vector of distances from planets to Sun   % meters
```

Problem Outputs: (with units)

Vector of forces in Newtons

Other variables: (with units)

none

Equation:

See above.

Algorithm: See **green** comments below.

Type the comments in the editor window and use as a guide for typing the MATLAB statements.

Table is created from the input vectors and the output vector.

Run output:

```
Gravitation force from sun to planets (Newtons):
Mass (kg)          Distant (m)         Force (N)
1.0e+24 *
0.330000000000000   0.000000000000058   0.013066744105882
4.870000000000000   0.000000000000108   0.055218551931284
5.970000000000001   0.000000000000150   0.035409656824259
```

Some rounding differences allowed.

Start your program file by typing the following into your empty editor window.

- Type yours/your group and other required information comments.
- Type the algorithm as given below as comments
  - to guide you when writing the MATLAB instructions
  - to do the tasks to solve the given problem.

  Below a comment type the MATLAB statement(s) do what the comment says.

```
% submitter's name, GROUP #
% other group members' names
% program file name
% due date of the assignment
% statement(s) about collaboration
% a short narrative about what the file does

clc, clear all
format long
format compact

% ***** CONSTANTS *****
% get the universal gravitational constant

% ***** INPUT *****
% get vector with Mercury, Venus, Earth masses
% get vector with Mercury, Venus, Earth distances
% get mass for sun

% ***** COMPUTE *****
% compute vector forces from sun to all planets
% create table from vectors: mass, distant, force

% ***** OUTPUT *****
% display table with title and column headers
```

NO error checking.
Do not use commands and statements beyond what has been taught on class.

New commands:
`format long, compact`
Compute with vectors.
Use a **dot** with operators as needed.
Build a table from vectors.
See text 5ed Ex 3.1 p. 71
`disp()` title
`disp()` column headers
`disp()` table
Use descriptive variables.

Submit via Canvas:

`assign02a.m` MATLAB script file
`assign02b.m` MATLAB script file

NOTE: Your submitted file(s) MUST be spelled and cased as instructed.

One submission per group. Canvas links members to files and rubric.