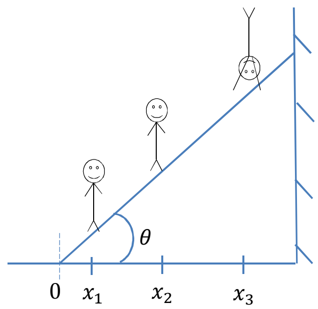
Name ____ section 2

Practice for Exam2. THIS IS NOT AN EXAM.

Complete Steps 7a, 7b, and 7c.

In order to take a cool picture, your team decides to have multiple people on the same ladder, of length L. (Off hand, it's already a bad idea.. but physics wise, it is possible???). The ladder is positioned as shown in the picture. Considering the mass (m_k) and position (x_k) of each individual, will it slip or not?



It is possible to calculate the minimum friction (unit less) required for **no** slippage. In other word, an actual friction greater than this value will make the picture possible:

$$f_{mini} = \frac{R_1}{L * \sin(\theta) * R_2}$$

where

$$R_1 = (m_1 * x_1 + m_2 * x_2 + .. + m_k * x_k) + m_{ladder} * \frac{L}{2} * \cos(\theta)$$

$$R_2 = (m_1 + m_2 + .. + m_k) + m_{ladder}$$

We will assume that the ladder is set on a surface that has an actual friction of 0.4. By comparing the calculated minimum friction with this one, it will be easy to determine if this project is smart…

USE ONLY THE KNOWLEDGE TAUGHT IN CLASS and IN THE LECTURE UP TO TODAY.

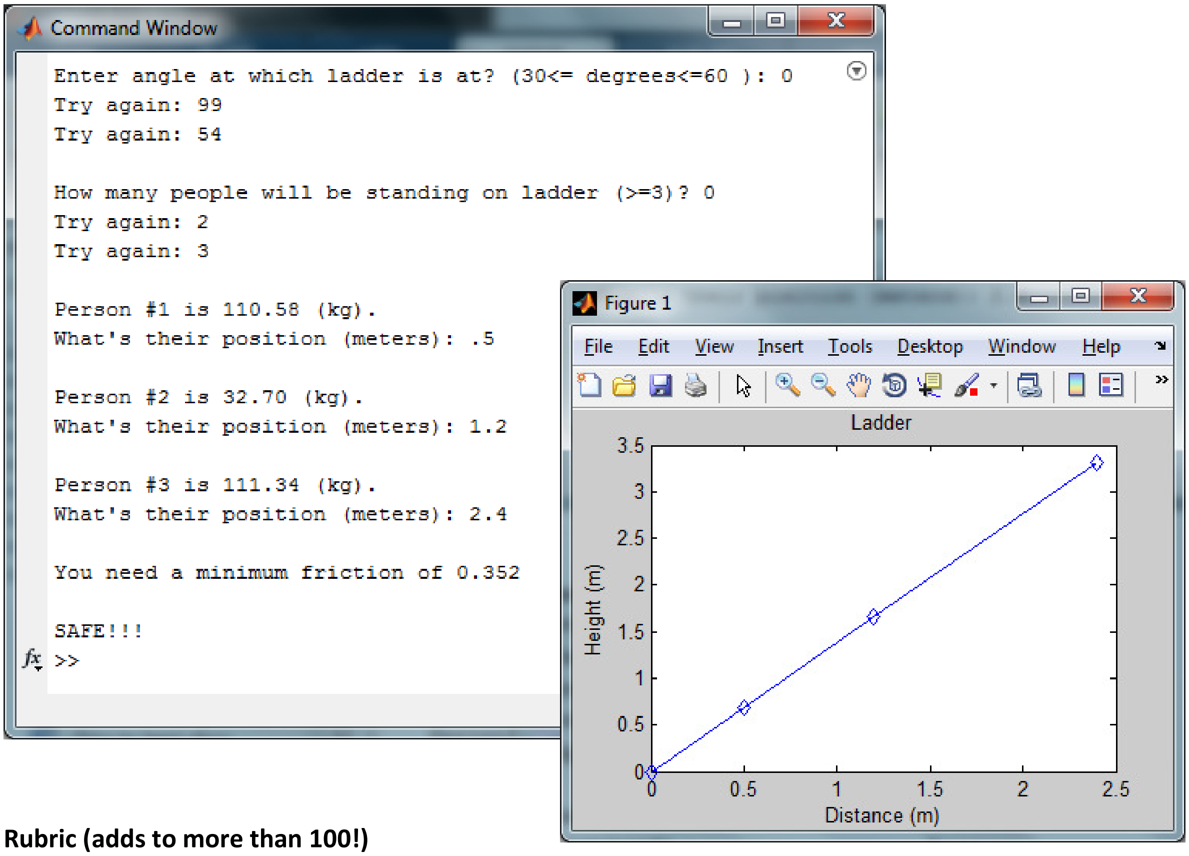
Requirements for the code:

- Hardcode the characteristics of the ladder in variables. It is 5meters long, and 18kg.
- **Prompt the user** for:
  - the angle at which the ladder is set at. This angle should be between 30 and 60 degrees both included. Trap the user.
  - the number of people on the ladder. Trap the user. This should be strictly greater than 2.
- Using the most appropriate loop, determine the characteristics of each person:
  - Generate a random mass for each person. That mass should be a float between 20 and 120kg (both excluded). Display the person's number and their mass. (see output page2)
  - Prompt for their horizontal x position (in meters). DO NOT TRAP. Assume correct values. (but when you test, remember the ladder is only 5 meters, so the maximum x is even smaller…)
  - The **same** loop shall be responsible for calculating R_1 and R_2 mentioned in the equation above, using running totals **only**. Notice that there is a portion of those equations that does NOT fall into a repeat pattern. Initialize your running totals to those values instead of zero, then let the loop do the part inside the parentheses!
  - Store each horizontal x-position in a vector, as they are necessary to plot the ladder in the end.
- Calculate/display the minimum friction required for no slippage. Show 3 decimals.
- Indicate if it is safe or not to proceed with taking the picture.
- Plot the ladder, and the people on it using a marker. The x values are already in a vector. However, you will need to insert the 0 in front (like lab12). **Then**, the y values are simply calculated using the slope equation:

  $$y(x) = \tan(\theta) * x$$

  Label properly as usual.

One example of when I run:



Rubric (adds to more than 100!)	
Intro	3
Clean up	3
Spacing of code & output	5
Indentation	5
Algorithm	5
Hardcode ladder data	2
Prompt angle	5
Trap angle	5
Prompt people	5
Trap people	5
Start running totals	5
Choose correct loop, write it well	5
Generate mass	5
Display person's number & mass	5
Prompt distance	5
Calculate R1	5
Calculate R2	5
Store in a vector	5
Calculate/display min friction	5
Safe/not safe	5
Vectors to plot	5
Plot ladder, labels etc..	5
Testing	5

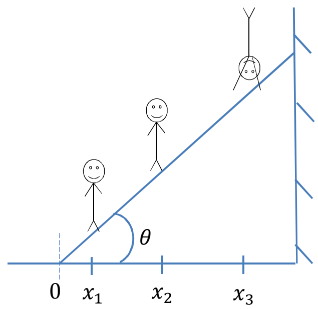
```matlab
%ladder safe or not?
%by Caroline
%section 2  TWO  only!

clc
clear
close all

%ladder data
myLength = 5 ;
massLadder = 18 ;

%prompt angle
myAngle = input('Enter angle at which ladder is at? (30<= degrees<=60 ): ');
while myAngle<30 || myAngle>60
    myAngle = input('Try again: ');
end

%prompt user for how many people
howManyPeople = input('\nHow many people will be standing on ladder (>=3)? ');
while howManyPeople<=2
    howManyPeople = input('Try again: ');
end

%set up running totals (initialize with ladder weight reactions)
reaction1 = massLadder*myLength/2*cosd(myAngle);
reaction2 = massLadder;

%loop to prompt each mass and distance
for k = 1:howManyPeople

    %generate mass
    mass = rand*100+20;

    %tell people number
    fprintf('\nPerson #%d is %.2f (kg).\n',k, mass);

    %prompt each distance
    x_dist(k) = input('What''s their position (meters): ');
    while x_dist(k)<=0 || x_dist(k)> myLength*cosd(myAngle)
        x_dist(k) = input('Ya... try between 0 and the projected length of the ladder.. duh: ');
    end

    %calculate reaction at bottom of ladder
    reaction2 = reaction2 + mass;

    %calculate reaction at top of ladder
    reaction1 = reaction1 + mass*x_dist(k);
end

%calculate minimum friction needed for no slippage
minFriction = reaction1/(myLength*sind(myAngle)*reaction2);
fprintf('\nYou need a minimum friction of %.3f\n',minFriction);

%decide if safe or not
if minFriction<.4
    fprintf('\nSAFE!!!\n');
else
    fprintf('\nDANGER!!!\n');
end

%plot ladder
xs = [0, x_dist];
ys = tand(myAngle)*xs;
plot(xs,ys,'d-');
title('Ladder');
xlabel('Distance (m)');
ylabel('Height (m)');

%see my cover sheet for testing.
```

Name ____ section 3

Practice for Exam2. THIS IS NOT AN EXAM.

Complete Steps 7a, 7b, and 7c.

In order to take a cool picture, your team decides to have multiple people on the same ladder, of length L. (Off hand, it's already a bad idea.. but physics wise, it is possible???). The ladder is positioned as shown in the picture. Considering the mass (m_k) and position (x_k) of each individual, will it slip or not?



It is possible to calculate the minimum friction (unit less) required for **no** slippage. In other word, an actual friction greater than this value will make the picture possible:

$$f_{mini} = \frac{R_1}{L * \sin(\theta) * R_2}$$

where

$$R_1 = (m_1 * x_1 + m_2 * x_2 + .. + m_k * x_k) + m_{ladder} * \frac{L}{2} * \cos(\theta)$$

$$R_2 = (m_1 + m_2 + .. + m_k) + m_{ladder}$$

We will assume that the ladder is set on a surface that has an actual friction of 0.4. By comparing the calculated minimum friction with this one, it will be easy to determine if this project is smart…

USE ONLY THE KNOWLEDGE TAUGHT IN CLASS and IN THE LECTURE UP TO TODAY.

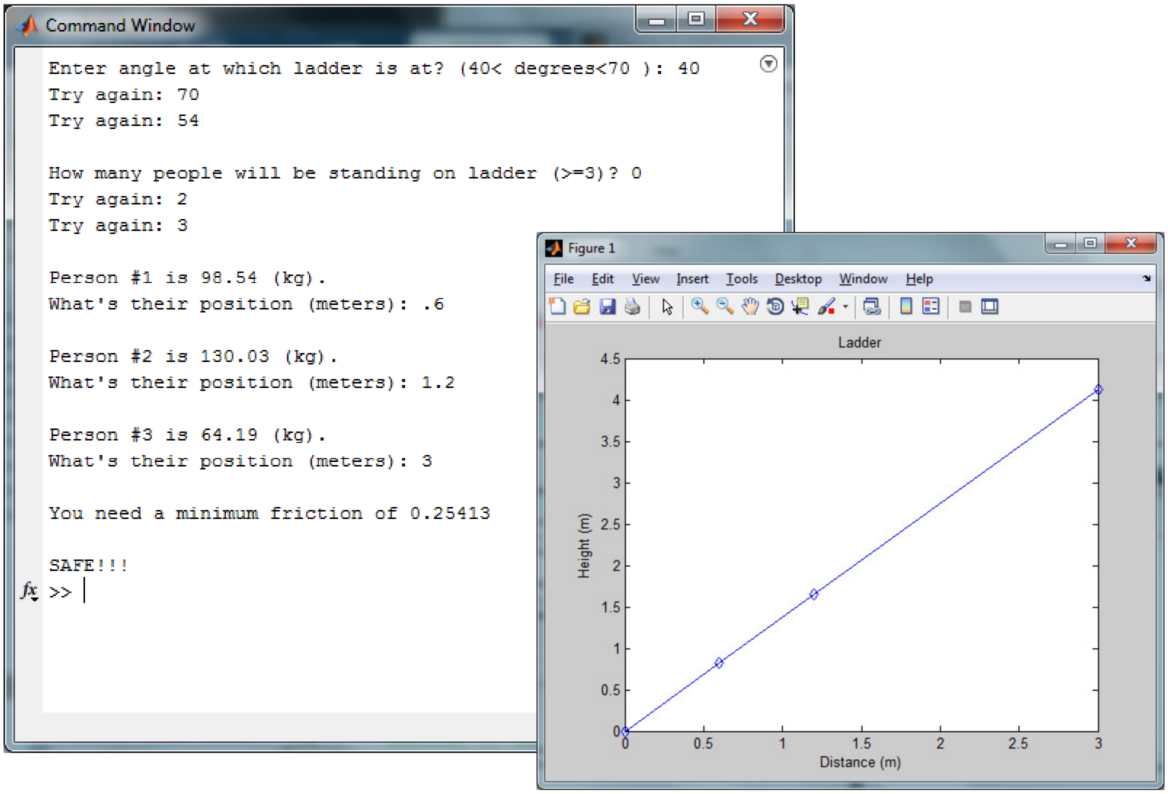
Requirements for the code:

- Hardcode the characteristics of the ladder in variables. It is 7meters long, and 22kg.
- **Prompt the user** for:
  - the angle at which the ladder is set at. This angle should be between 40 and 70 degrees both excluded. Trap the user.
  - the number of people on the ladder. Trap the user. This should be strictly greater than 2.
- Using the most appropriate loop, determine the characteristics of each person:
  - Generate a random mass for each person. That mass should be a float between 50 and 150kg (both excluded). Display the person's number and their mass. (see output page2)
  - Prompt for their horizontal x position (in meters). DO NOT TRAP. Assume correct values. (but when you test, remember the ladder is only 7 meters, so the maximum x is even smaller…)
  - The **same** loop shall be responsible for calculating R_1 and R_2 mentioned in the equation above, using running totals **only**. Notice that there is a portion of those equations that does NOT fall into a repeat pattern. Initialize your running totals to those values instead of zero, then let the loop do the part inside the parentheses!
  - Store each horizontal x-position in a vector, as they are necessary to plot the ladder in the end.
- Calculate/display the minimum friction required for no slippage. Show 5 decimals.
- Indicate if it is safe or not to proceed with taking the picture.
- Plot the ladder, and the people on it using a marker. The x values are already in a vector. However, you will need to insert the 0 in front (like lab12). **Then**, the y values are simply calculated using the slope equation:

  $$y(x) = \tan(\theta) * x$$

  Label properly as usual.

One example of when I run:



Rubric (adds to more than 100!)

Intro	3
Clean up	3
Spacing of code & output	5
Indentation	5
Algorithm	5
Hardcode ladder data	2
Prompt angle	5
Trap angle	5
Prompt people	5
Trap people	5
Start running totals	5
Choose correct loop, write it well	5
Generate mass	5
Display person's number & mass	5
Prompt distance	5
Calculate R1	5
Calculate R2	5
Store in a vector	5
Calculate/display min friction	5
Safe/not safe	5
Vectors to plot	5
Plot ladder, labels etc..	5
Testing	5

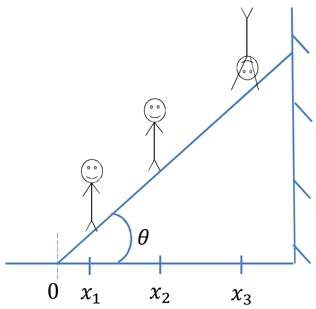
```matlab
%ladder safe or not?
%by Caroline
%section 3

clc
clear
close all

%ladder data
myLength = 7 ;
massLadder = 22 ;

%prompt angle
myAngle = input('Enter angle at which ladder is at? (40< degrees<70 ): ');
while myAngle<=40 || myAngle>=70
    myAngle = input('Try again: ');
end

%prompt user for how many people
howManyPeople = input('\nHow many people will be standing on ladder (>=3)? ');
while howManyPeople<=2
    howManyPeople = input('Try again: ');
end

%set up running totals (initialize with ladder weight reactions)
reaction1 = massLadder*myLength/2*cosd(myAngle);
reaction2 = massLadder;

%loop to prompt each mass and distance
for k = 1:howManyPeople

    %generate mass
    mass = rand*100+50;

    %tell people number
    fprintf('\nPerson #%d is %.2f (kg).\n',k, mass);

    %prompt each distance
    x_dist(k) = input('What''s their position (meters): ');
    while x_dist(k)<=0 || x_dist(k)> myLength*cosd(myAngle)
        x_dist(k) = input('Ya... try between 0 and the projected length of the ladder.. duh: ');
    end

    %calculate reaction at bottom of ladder
    reaction2 = reaction2 + mass;

    %calculate reaction at top of ladder
    reaction1 = reaction1 + mass*x_dist(k);
end

%calculate minimum friction needed for no slippage
minFriction = reaction1/(myLength*sind(myAngle)*reaction2);
fprintf('\nYou need a minimum friction of %.5f\n',minFriction);

%decide if safe or not
if minFriction<.4
    fprintf('\nSAFE!!!\n');
else
    fprintf('\nDANGER!!!\n');
end

%plot ladder
xs = [0, x_dist];
ys = tand(myAngle)*xs;
plot(xs,ys,'d-');
title('Ladder');
xlabel('Distance (m)');
ylabel('Height (m)');

%see my cover sheet for testing.
```

Name ____ section 4

Practice for Exam2. THIS IS NOT AN EXAM.

Complete Steps 7a, 7b, and 7c.

In order to take a cool picture, your team decides to have multiple people on the same ladder, of length L. (Off hand, it's already a bad idea.. but physics wise, it is possible???). The ladder is positioned as shown in the picture. Considering the mass (m_k) and position (x_k) of each individual, will it slip or not?



It is possible to calculate the minimum friction (unit less) required for **no** slippage. In other word, an actual friction greater than this value will make the picture possible:

$$f_{mini} = \frac{R_1}{L * \sin(\theta) * R_2}$$

where

$$R_1 = (m_1 * x_1 + m_2 * x_2 + .. + m_k * x_k) + m_{ladder} * \frac{L}{2} * \cos(\theta)$$

$$R_2 = (m_1 + m_2 + .. + m_k) + m_{ladder}$$

We will assume that the ladder is set on a surface that has an actual friction of 0.4. By comparing the calculated minimum friction with this one, it will be easy to determine if this project is smart…

USE ONLY THE KNOWLEDGE TAUGHT IN CLASS and IN THE LECTURE UP TO TODAY.

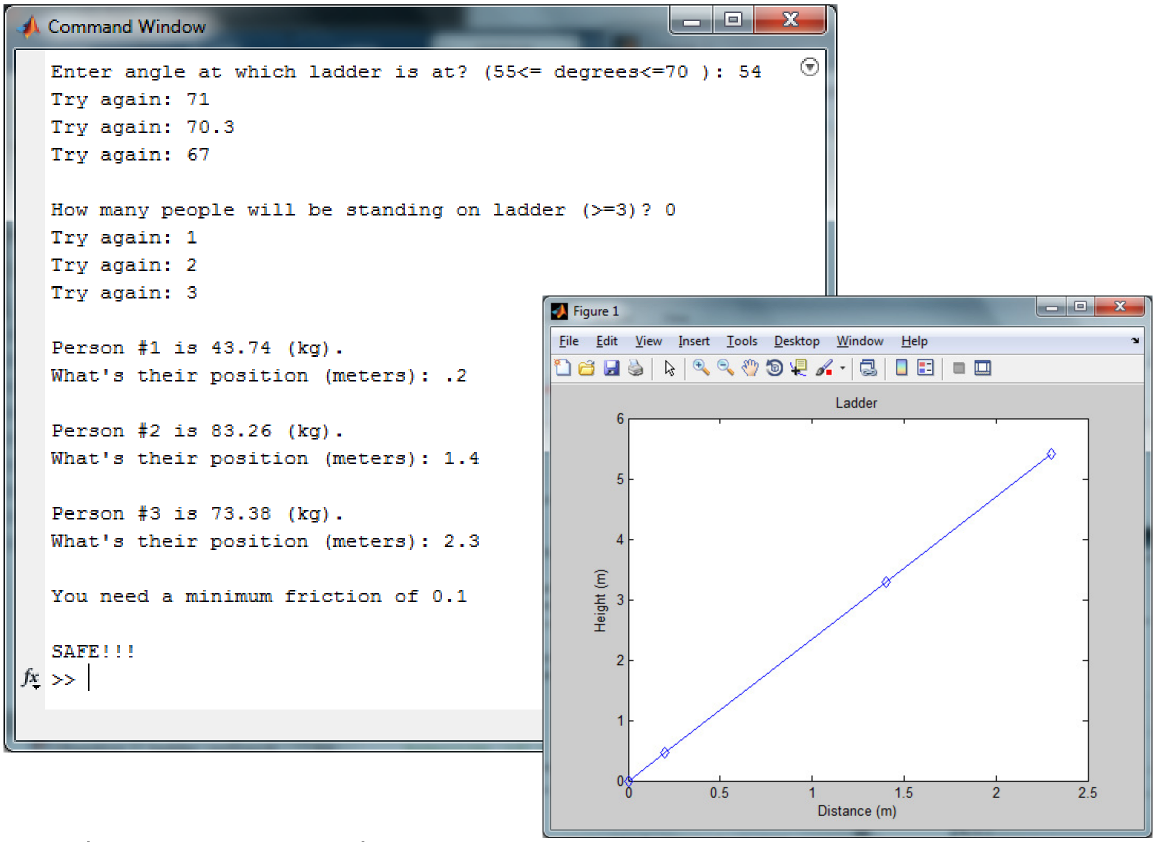
Requirements for the code:

- Hardcode the characteristics of the ladder in variables. It is 12meters long, and 33kg.
- **Prompt the user** for:
  - the angle at which the ladder is set at. This angle should be between 55 and 70 degrees both included. Trap the user.
  - the number of people on the ladder. Trap the user. This should be strictly greater than 2.
- Using the most appropriate loop, determine the characteristics of each person:
  - Generate a random mass for each person. That mass should be a float between 10 and 90kg (both excluded). Display the person's number and their mass. (see output page2)
  - Prompt for their horizontal x position (in meters). DO NOT TRAP. Assume correct values. (but when you test, remember the ladder is only 12 meters, so the maximum x is even smaller…)
  - The **same** loop shall be responsible for calculating R_1 and R_2 mentioned in the equation above, using running totals **only**. Notice that there is a portion of those equations that does NOT fall into a repeat pattern. Initialize your running totals to those values instead of zero, then let the loop do the part inside the parentheses!
  - Store the horizontal x-position in a vector, as they are necessary to plot the ladder in the end.
- Calculate/display the minimum friction required for no slippage. Show 1 decimal.
- Indicate if it is safe or not to proceed with taking the picture.
- Plot the ladder, and the people on it using a marker. The x values are already in a vector. However, you will need to insert the 0 in front (like lab12). **Then**, the y values are simply calculated using the slope equation:

  $$y(x) = \tan(\theta) * x$$

  Label properly as usual.

One example of when I run:



Rubric (adds to more than 100!)

Intro	3
Clean up	3
Spacing of code & output	5
Indentation	5
Algorithm	5
Hardcode ladder data	2
Prompt angle	5
Trap angle	5
Prompt people	5
Trap people	5
Start running totals	5
Choose correct loop, write it well	5
Generate mass	5
Display person's number & mass	5
Prompt distance	5
Calculate R1	5
Calculate R2	5
Store in a vector	5
Calculate/display min friction	5
Safe/not safe	5
Vectors to plot	5
Plot ladder, labels etc..	5
Testing	5

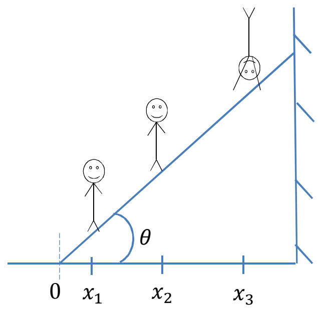
```matlab
%ladder safe or not?
%by Caroline
%section 4 FOUR only

clc
clear
close all

%ladder data
myLength = 12 ;
massLadder = 33 ;

%prompt angle
myAngle = input('Enter angle at which ladder is at? (55<= degrees<=70 ): ');
while myAngle<55 || myAngle>70
    myAngle = input('Try again: ');
end

%prompt user for how many people
howManyPeople = input('\nHow many people will be standing on ladder (>=3)? ');
while howManyPeople<=2
    howManyPeople = input('Try again: ');
end

%set up running totals (initialize with ladder weight reactions)
reaction1 = massLadder*myLength/2*cosd(myAngle);
reaction2 = massLadder;

%loop to prompt each mass and distance
for k = 1:howManyPeople

    %generate mass
    mass = rand*80+10;

    %tell people number
    fprintf('\nPerson #%d is %.2f (kg).\n',k, mass);

    %prompt each distance
    x_dist(k) = input('What''s their position (meters): ');
    while x_dist(k)<=0 || x_dist(k)> myLength*cosd(myAngle)
        x_dist(k) = input('Ya... try between 0 and the projected length of the ladder.. duh: ');
    end

    %calculate reaction at bottom of ladder
    reaction2 = reaction2 + mass;

    %calculate reaction at top of ladder
    reaction1 = reaction1 + mass*x_dist(k);
end

%calculate minimum friction needed for no slippage
minFriction = reaction1/(myLength*sind(myAngle)*reaction2);
fprintf('\nYou need a minimum friction of %.1f\n',minFriction);

%decide if safe or not
if minFriction<.4
    fprintf('\nSAFE!!!\n');
else
    fprintf('\nDANGER!!!\n');
end

%plot ladder
xs = [0, x_dist];
ys = tand(myAngle)*xs;
plot(xs,ys,'d-');
title('Ladder');
xlabel('Distance (m)');
ylabel('Height (m)');

%see my cover sheet for testing.
```

Name ____ section 5

Practice for Exam2. THIS IS NOT AN EXAM.

Complete Steps 7a, 7b, and 7c.

In order to take a cool picture, your team decides to have multiple people on the same ladder, of length L. (Off hand, it's already a bad idea.. but physics wise, it is possible???). The ladder is positioned as shown in the picture. Considering the mass (m_k) and position (x_k) of each individual, will it slip or not?



It is possible to calculate the minimum friction (unit less) required for **no** slippage. In other word, an actual friction greater than this value will make the picture possible:

$$f_{mini} = \frac{R_1}{L * \sin(\theta) * R_2}$$

where

$$R_1 = (m_1 * x_1 + m_2 * x_2 + .. + m_k * x_k) + m_{ladder} * \frac{L}{2} * \cos(\theta)$$

$$R_2 = (m_1 + m_2 + .. + m_k) + m_{ladder}$$

We will assume that the ladder is set on a surface that has an actual friction of 0.4. By comparing the calculated minimum friction with this one, it will be easy to determine if this project is smart…

USE ONLY THE KNOWLEDGE TAUGHT IN CLASS and IN THE LECTURE UP TO TODAY.

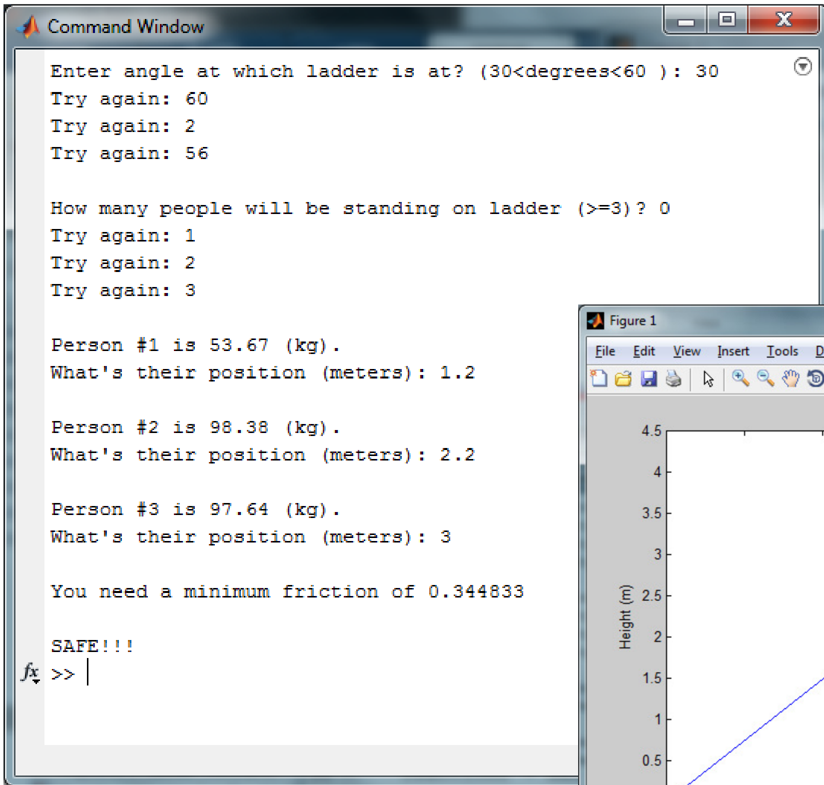
Requirements for the code:

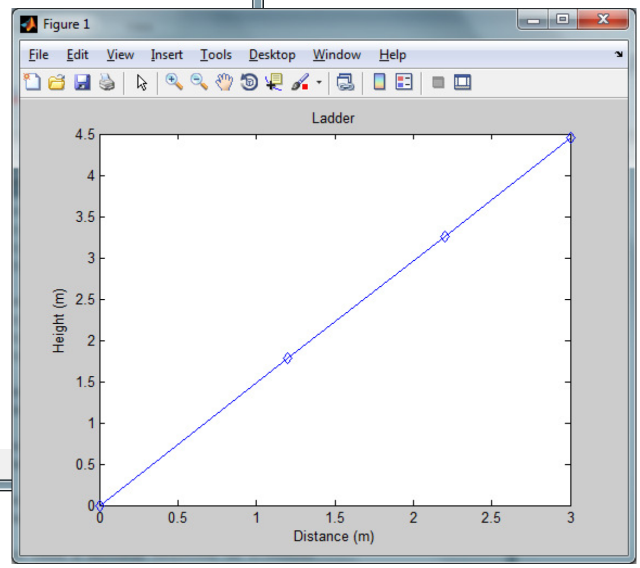
- Hardcode the characteristics of the ladder in variables. It is 8meters long, and 54kg.
- **Prompt the user** for:
  - the angle at which the ladder is set at. This angle should be between 30 and 60 degrees both excluded. Trap the user.
  - the number of people on the ladder. Trap the user. This should be strictly greater than 2.
- Using the most appropriate loop, determine the characteristics of each person:
  - Generate a random mass for each person. That mass should be a float between 45 and 100kg (both excluded). Display the person's number and their mass. (see output page2)
  - Prompt for their horizontal x position (in meters). DO NOT TRAP. Assume correct values. (but when you test, remember the ladder is only 8 meters, so the maximum x is even smaller…)
  - The **same** loop shall be responsible for calculating R_1 and R_2 mentioned in the equation above, using running totals **only**. Notice that there is a portion of those equations that does NOT fall into a repeat pattern. Initialize your running totals to those values instead of zero, then let the loop do the part inside the parentheses!
  - Store the horizontal x-position in a vector, as they are necessary to plot the ladder in the end.
- Calculate/display the minimum friction required for no slippage. Show 6 decimals.
- Indicate if it is safe or not to proceed with taking the picture.
- Plot the ladder, and the people on it using a marker. The x values are already in a vector. However, you will need to insert the 0 in front (like lab12). **Then**, the y values are simply calculated using the slope equation:

$$y(x) = \tan(\theta) * x$$

  Label properly as usual.

One example of when I run:





Rubric (adds to more than 100!)

Intro	3
Clean up	3
Spacing of code & output	5
Indentation	5
Algorithm	5
Hardcode ladder data	2
Prompt angle	5
Trap angle	5
Prompt people	5
Trap people	5
Start running totals	5
Choose correct loop, write it well	5
Generate mass	5
Display person's number & mass	5
Prompt distance	5
Calculate R1	5
Calculate R2	5
Store in a vector	5
Calculate/display min friction	5
Safe/not safe	5
Vectors to plot	5
Plot ladder, labels etc..	5
Testing	5

```
%ladder safe or not?
%by Caroline
%section 5, FIVE only!

clc
clear
close all

%ladder data
myLength = 8 ;
massLadder = 54 ;

%prompt angle
myAngle = input('Enter angle at which ladder is at? (30<degrees<60 ): ');
while myAngle<=30 || myAngle>=60
    myAngle = input('Try again: ');
end

%prompt user for how many people
howManyPeople = input('\nHow many people will be standing on ladder (>=3)? ');
while howManyPeople<=2
    howManyPeople = input('Try again: ');
end

%set up running totals (initialize with ladder weight reactions)
reaction1 = massLadder*myLength/2*cosd(myAngle);
reaction2 = massLadder;

%loop to prompt each mass and distance
for k = 1:howManyPeople

    %generate mass
    mass = rand*55+45;

    %tell people number
    fprintf('\nPerson #%d is %.2f (kg).\n',k, mass);

    %prompt each distance
    x_dist(k) = input('What''s their position (meters): ');
    while x_dist(k)<=0 || x_dist(k)> myLength*cosd(myAngle)
        x_dist(k) = input('Ya... try between 0 and the projected length of the ladder.. duh: ');
    end

    %calculate reaction at bottom of ladder
    reaction2 = reaction2 + mass;

    %calculate reaction at top of ladder
    reaction1 = reaction1 + mass*x_dist(k);
end

%calculate minimum friction needed for no slippage
minFriction = reaction1/(myLength*sind(myAngle)*reaction2);
fprintf('\nYou need a minimum friction of %f\n',minFriction);

%decide if safe or not
if minFriction<.4
    fprintf('\nSAFE!!!\n');
else
    fprintf('\nDANGER!!!\n');
end

%plot ladder
xs = [0, x_dist];
ys = tand(myAngle)*xs;
plot(xs,ys,'d-');
title('Ladder');
xlabel('Distance (m)');
ylabel('Height (m)');

%see my cover sheet for testing.
```