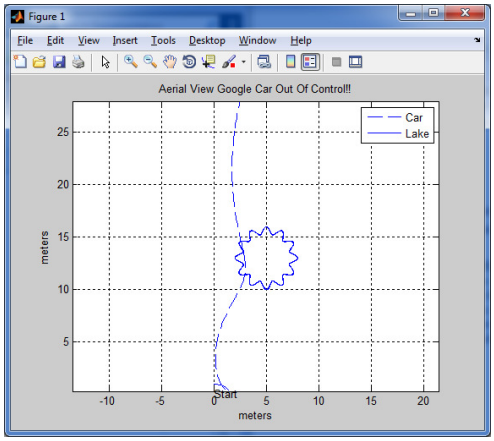
Name: _____ SECTION: _____ (CRUCIAL. ALL MIXED!!!)
Practice for exam1. Spring 2016. VersionA.



A Google® car has been hijacked due to software bugs and security hacks. Oh no! Thankfully, you've hacked yourself in the system and are trying to save the car from ending up in a lake!

Note: all this is entirely made up. ☺ Don't try to figure physics out of all this, just fun math and plotting.

I DO NOT ANSWER QUESTIONS RELATED TO THE PROBLEM OR THE CODE. YOU CAN TRY STUFF, SEE IF IT WORKS, OR DOESN'T. DO YOUR BEST. COMMENT WHEN IT CRASHES. NO CODE SHOULD CRASH.
DO NOT STAY STUCK. CODE EVERYTHING RELATED TO THE PLOT FIRST AND COMMENT IT OUT. I CAN GIVE POINTS THAT WAY.

The car

As the hacker, you are able to override two **given** coefficients: $k_{outOfControl}$ and $maxTime$. Create a variable for each so you can change those easily. Both are scalars.
To plot the car's trajectory, its (x,y) coordinates are defined by the following equations that are dependent on time:

$$x(t) = t + \sin(k_{outOfControl} * t)$$
$$y(t) = maxTime * t^2 + \cos(k_{outOfControl} * t)$$

You will need a lot of values in order the plot the car trajectory. Therefore, define a lot of values for t, from 0 to $maxTime$ (pick at least 200values). x and y are both in meters.

The lake

While you don't control the lake as much, you **know** its center location x_{center} and y_{center} and another coefficient k_{lake}. Again create a variable for each so you can change their value easily. They're all scalars.
The equations to plot the lake *y* vs. *x* are:

$$x(\beta) = k_{lake} * \left(\sin^2(k_{lake} * \beta) + \cos^4(k_{lake} * \beta)\right) * \sin(\beta) + x_{center}$$
$$y(\beta) = k_{lake} * \left(\sin^2(k_{lake} * \beta) + \cos^4(k_{lake} * \beta)\right) * \cos(\beta) + y_{center}$$

Where β is once again a lot of angles, pre-set from 0 to 10π. x and y are both in meters.

Your goal is to plot both the car path (y vs. x) and the lake (y vs. x) on the same figure, and play with each given to save the car from going into/through the lake. Fill in the blanks in the table below by reusing your code over and over:

$k_{outOfControl}$	$maxTime$	x_{center}	y_{center}	k_{lake}	Car goes into (through) lake? Answer YES or NO if blank
4	3	5	13	4	…
…	3	5	13	**3**	NO
12	…	**3**	**10**	**2.5**	NO

The engineering process is partially completed.
Fill in Step1 yourself: (5pts)

Givens: k out of control, max time, xcenter, ycenter, k of lake
Find: scenario so car does not go through lake!

Step2: plot already done. See front of cover sheet.

Step3: ALL equations in front of the cover sheet. No need for additional ones.

Step4: no assumptions can be made to simplify the problem.

Step5/Step6: mmm.. good luck doing this analytically. NOT TODAY!

Step7: ALL of step7 is done in the code as usual. The algorithm should be your comments as usual.

There is no output in this code. Suppress the output of absolutely all calculations. NO proof of testing possible!

Format the plot fully as shown in the figure in the front page. Make sure to use different line types to differentiate lake from car. The rest is up to you. Indicate the word 'Start' at the location (0,0),

Bonus: indicate the word 'Lake' at the center location of the lake!

Rubric (100pts):

Step1 filled in	4pts	
Table completed	2pts	
In script:		
Name, section, description		3pts
All clean up commands		3pts
Spacing of code – skip lines		5pts
No indenting of any kind		5pts
Comments/algorithm		10pts
Good variables names		5pts
Variables hardcoded/easy to change		5pts
Vectors to plot the car		5+5+5pts
Vectors to plot the lake		5+5+5pts
Correct use of element per element operator		5pts
No extra/useless parentheses		5pts
Proper use of semi-colons		5pts
Proper plot command/setup		5pts
All plot formatting		8pts

5pts penalty if code crashes.

Bonus:

What is the difference between binary language and machine language? An example is ok.

Binary language: 0 and 1's
Machine language started adding words: add a b; for example to add a+b

```
%{
Practice exam1. Google cars
Prof.Liron
Sp16
%}

clc;
clear;
close all;

%define path of car
xsCar =
ysCar =

%lake location

%lake data
xsPond =
ysPond =

%plot and formatting
plot(xsCar,ysCar,'--r',xsPond,ysPond,'b');
text(0,0,'Start');
axis equal;
grid on;
title('Aerial View Google Car Out Of Control!!');
legend('Car','Lake');
xlabel('meters');
ylabel('meters');

%bonus
text(,,'Lake');

%no testing. nothing shows up! answer cover sheet!
```

```
%{
Practice exam1. Google cars
Prof.Liron
Sp16
%}

clc;
clear;
close all;

%define path of car
coeffControl = 12;
maxTime = 2;
timesCar = linspace(0,maxTime,200);
xsCar = timesCar+sin(coeffControl*timesCar);
ysCar = maxTime*timesCar.^2+cos(coeffControl*timesCar);

%lake location
xCenterPond = 3; %meters
yCenterPond = 10;

%lake data
kPond = 2.5;
thetas = linspace(0,10*pi,500);
xsPond = kPond*(sin(kPond*thetas).^2+cos(kPond*thetas).^4).*sin(thetas)+xCenterPond; %↙
SINE
ysPond = kPond*(sin(kPond*thetas).^2+cos(kPond*thetas).^4).*cos(thetas)+yCenterPond; %↙
COSINE

%plot and formatting
plot(xsCar,ysCar,'--r',xsPond,ysPond,'b');
text(0,0,'Start');
axis equal;
grid on;
title('Aerial View Google Car Out Of Control!!');
legend('Car','Lake');
xlabel('meters');
ylabel('meters');

%bonus
text(xCenterPond,yCenterPond,'Lake');

%no testing. nothing shows up! answer cover sheet!
```