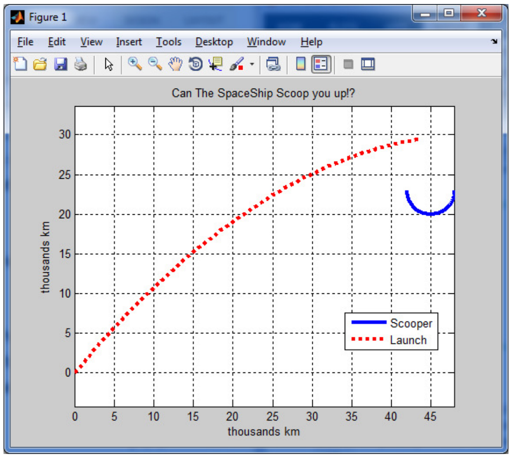
Name: ______________ANSWER__________________________ SECTION: _____ (CRUCIAL. ALL MIXED!!!)
Practice for exam1. Spring 2016. VersionB.



Well, you've found your sensor on Mars and you've found a way to launch yourself at a given angle and velocity. However, your "fuel time" is limited. Hopefully, by the time you run out of fuel, you'll be above the gigantic scooper NASA set up in space, so you can just "fall" back into it!

Note: all this is entirely made up. ☺ Don't try to figure physics out of all this, just fun math and plotting. Even the units are far-stretched. Consider the distances in thousands of kilometers. ☺

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 scooper

The scooper is a half circle placed in space. As a NASA scientist, you can control the following **givens**: radius R, x_{center} and y_{center} points. Create a variable for each so you can change those easily. All are scalars.
To plot it, the scooper's (x,y) coordinates are defined by the following equations that are dependent on angles:

$$x(\emptyset) = R * \cos(\emptyset) + x_{center}$$
$$y(\emptyset) = R * \sin(\emptyset) + y_{center}$$

Since the scooper is only a half circle, use a lot of angles going from 180° to 360°. (Feel free to use π to 2π instead).

The astronaut's trajectory

You know how much time the fuel will last ($t_{\max}$), a launch angle β in degrees, and an initial velocity v_o you'll take off at (don't worry about units too much). Again, create variables so you can change their values easily. All are scalars.
The equations to plot are based on time t which needs to go from zero to $t_{\max}$ so you can plot the trajectory:

$$x(t) = v_o * \cos(\beta) * t$$
$$y(t) = -1.1\, t^2 + v_o * \sin(\beta) * t$$

Your goal is to plot both the scooper (y vs. x) and the astronaut (y vs. x) on the same figure, and play with each given to save the astronaut by setting the max fuel time perfectly so all engines cut off anywhere **above** the scooper! Fill in the blanks in the table below by reusing your code over and over:

Radius: R	x_{center}	y_{center}	Fuel time: $t_{\max}$	Launch angle: β	v_o	Are you right above the scooper when all engines are off? Answer YES or NO if blank
3	45	23	4.5	50	15	……
5	**20**	23	…….	**80**	…….	YES
50	……	……	**40**	**15**	**22.5**	YES

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

Givens: radius R, xcenter, ycenter, tmax, angle beta, and intial velocity
Find: a plausible scenario that saves the astronaut, or whether the astronaut is dead.

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 need to 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 scooper from trajectory.

In the legend() command, the LAST TWO words before the end should be as: `………, 'location', 'best');`
This will make sure the box isn't on top of your trajectory if possible.

Bonus: indicate the word 'Landing Site' at the 'center' location of the scooper!

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 scooper	5+5+5pts
	Vectors to plot the astronaut	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. Comment stuff out!

Bonus:

What is the difference between operating system and an application?

OS is the brain of the computer.
Applications are programs you download to use: MATLAB, CATIA, … "little" things like that.

```matlab
%{
can the space ship scoop you up?
by Prof.Liron
practice exam1
%}

clc;
clear;
close all;

%define scoop data


%define launch data


%plot
plot(xscoop,yscoop, xlaunch, ylaunch,':r'); %or hold on; method.

%plot formatting
title('Can The SpaceShip Scoop you up!?');
xlabel('thousands km');
ylabel('thousands km');
legend('Scooper','Launch');
grid on;
axis equal;

%bonus
%text( , ,'Landing Site');
```

```
%{
can the space ship scoop you up?
by Prof.Liron
practice exam1
%}

clc;
clear;
close all;

%define scoop data
xCenterScoop = 45;
yCenterScoop = 23;
radiusScoop = 3;
angles = linspace(180,360);  %you can do pi, to 2pi
xscoop = radiusScoop*cosd(angles)+xCenterScoop;   %then you don't use 'd'
yscoop = radiusScoop*sind(angles)+yCenterScoop;

%define launch data
fuelTime = 4.5; %hrs
angLaunch = 50; %degrees
velo = 15;
times = linspace(0,fuelTime);  %caution: time() is a function. Press F1 to know
xlaunch = velo*cosd(angLaunch)*times;
ylaunch = velo*sind(angLaunch)*times - 1.1*times.^2;

%plot
plot(xscoop,yscoop, xlaunch, ylaunch,':r','linewidth',3); %or hold on; method

%plot formatting
title('Can The SpaceShip Scoop you up!?');
xlabel('thousands km');
ylabel('thousands km');
legend('Scooper','Launch','location','best');
grid on;
axis equal;

%bonus
text(xCenterScoop,yCenterScoop,'Landing Site');
```