levelset[e_,k_]:=Module[{k1,set1},k1=k;set1=ContourPlot[e==k1,{x,-8,8},{y,-8,8},Axes→True];Return[set1]]
C. Place the answer to a question right after it.
D. Remember to use alt + 7 to format cells in which you will be typing text. You don’ t need to format cells in which you will be writing Mathematica code.
To submit
E. Save the notebook as *yourname_assignment 1.nb*
F. Evaluate entire notebook.
G. Save as a .pdf file.
H. Submit to Canvas These are due on Friday, March 11.
In new cells, experiment by typing and evaluating the following:

A. Solve[2x=7]
B. a=9; Solve[2x==a] (that’s a semicolon in there)
C. a=9 [Return] Solve[2x==a] (the carriage return is not Shift+Return).

Sum up what you have deduced about semi-colons, the Solve command, and use of equal signs. In particular, what’s the difference between B and C?
In new cells, experiment with this by typing and evaluating the following:

C. Solve [3x==y]
D. Solve[3x==y,{x}].

Sum up what you have deduced about the Solve command and solving for particular variables.
What are the pros and cons of using NSolve instead of Solve?
Play around with the ContourPlot command: use ContourPlot to draw a parabola, a hyperbola, a line, and a cubic, some with and some without axes.
Change the bounds
on

```
Show[ContourPlot[(x - 1)^2 + (y - 7)^2 == 10, {x, -3, 5}, {y, 0, 12}, Axes → True],
 ContourPlot[y == 3 x^2 + 1, {x, -3, 5}, {y, 0, 12}, Axes → True]]
```

so that the circle comes out looking circular. (You’ll need to cut-and-paste it into a cell that isn’t formatted for text.) Briefly sum up what you think is happening.
Get *Mathematica* to draw a picture of a parabola and a line tangent to the parabola at some point other than the vertex. If you want to show off, include a (visible) point of tangency in your picture. (You may need to look up the command “Point” in the documentation under the Help tab.)
Play a little. Is it the case that every level set for f(x,y)=3xy+2y is a hyperbola? Explain.

Change the input expression so that the corresponding level set is a parabola.

How about two lines intersecting lines? Two parallel lines? The goal here is to only use one levelset command.

Next change the module itself. Write a module called levelsetwithbounds that takes as input an expression *e*, a value *k* and a number *M* and has as output the level set of $f(x, y) = e$ corresponding to the level *k* between the ranges $-M \leq x, y \leq M$. (Our levelset module only draws the picture where M=8.)

Use your module to draw a picture of part of parabola containing the vertex where the vertex is more than 40 away from the origin.
For the next question, you need the following module. Be sure to evaluate it.
log[n_]:=Module[{index},
If[n≤ 0,Return["Please enter a positive number"],Return[Log10[n]]]
(*This is the If command. It looks like this:
If[_____,______,______] where the first entry is a condition and the second and third entries are actions. *)
(* It works like this:
If the condition in the first entry holds, it does the thing in the second entry. If the condition doesn't hold, it does the thing in the third entry.*)
]
Evaluate the module. Then enter and evaluate the following:
log[100]
log[100000]
log[13.4]
log[-6]
Explain your results.
Plug a bunch of different kinds of numbers into the module below (after evaluating it).
ilikeevennumbers[n_]:=Module[{a},a=RandomInteger[2];
If[EvenQ[n],Return[{"Yippee!","Wonderful","Fantastic"}[[a+1]]]]
]
Explain what the module does.
Write a module that takes as input a positive integer *m* prints the first *m* positive odd numbers and returns the square root of the sum of the numbers it printed. You should use the Print command in the module. Look it up in the Wolfram Documentation center available under the Help tab. Run your module for 3 different numbers between 20 and 50. Do you have any thoughts about the output?

```
Solve[2 x = 7]
a = 9; Solve[2 x == a]
a = 9[Return] Solve[2 x == a]
```

```
Set::write : Tag Times in 2 x is Protected. >>

Solve::naqs : 7 is not a quantified system of equations and inequalities. >>

Solve[7]

{{x → 9/2}}

{{9[Return] (x → 9/2)}}
```

I belive that the semi-colon is used to supress output so that a variable can be declared and an expression contanining said variable can be evaluated on the same line. The solve command seems pretty self explanitory, it evaluates an equation. Equal signs on their own appear to be declaring a variable whereas double equal signs denote an equality to be evaluated. In the second bit of code we are evaluating expressions where a variable is defined, in one the variable is declared by a single = sign and supressed by a semicolen, in the next the variable is returned in addition to the solution to the equality.

```
Solve[3 x == y]
Solve[3 x == y, {x}]
```

```
{{y → 3 x}}
```

$$\left\{\left\{x \to \frac{y}{3}\right\}\right\}$$

My deductions about the Solve command have not changed since the problem I answered one minute ago. However, I have learned that including in the solve brackets a comma followed by one of the variables in squiggly brackets appears to be the syntax for telling Mathematica which variable you would like it to evaluate the expression for.

```
Solve[x^2 == 5]
NSolve[x^2 == 5]
```

$$\left\{\left\{x \to -\sqrt{5}\right\}, \left\{x \to \sqrt{5}\right\}\right\}$$

```
{{x → -2.23607}, {x → 2.23607}}
```

```
NSolve[y == 3 x^2 + 1 && (x - 1)^2 + (y - 7)^2 == 10, {x, y}]
Solve[y == 3 x^2 + 1 && (x - 1)^2 + (y - 7)^2 == 10, {x, y}]
```

```
{{x → -1.61082, y → 8.78426}, {x → 1.73922, y → 10.0747},
 {x → -1.10099, y → 4.63657}, {x → 0.972598, y → 3.83784}}
```

$$\left\{\left\{x \to -\frac{1}{6}\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}\right)\right)} - \frac{1}{2}\sqrt{\left(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3} - 4\Big/\left(\sqrt{\left(3\left(70+\frac{4141}{\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right)\right)},\right.\right.$$

$$y \to \frac{41}{6} - 1\Big/\left(\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right) + \frac{1}{2}\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,i\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}$$

$$\sqrt{\Bigg(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}-}$$

$$4\Big/\Bigg(\sqrt{\Bigg(3\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)\Bigg)\Big\},$$

$$\Big\{x\to-\frac{1}{6}\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}+$$

$$\frac{1}{2}\sqrt{\Bigg(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}-}$$

$$4\Big/\Bigg(\sqrt{\Bigg(3\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)\Bigg),$$

$$y\to\frac{41}{6}-1\Big/\Bigg(\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)-$$

$$\frac{1}{2}\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}$$

$$\sqrt{\Bigg(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}-}$$

$$4\Big/\Bigg(\sqrt{\Bigg(3\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)\Bigg)\Big\},$$

$$\Big\{x\to\frac{1}{6}\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}-$$

$$\frac{1}{2}\sqrt{\Bigg(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}+}$$

$$4\Big/\Bigg(\sqrt{\Bigg(3\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)\Bigg),$$

$$y\to\frac{41}{6}+1\Big/\Bigg(\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}\Bigg)-$$

$$\frac{1}{2}\sqrt{\Bigg(\frac{1}{3}\Bigg(70+\frac{4141}{\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathbb{i}\,\sqrt{4\,394\,085}\right)^{1/3}\Bigg)\Bigg)}$$

$$\sqrt{\left(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}+\right.}$$

$$\left.\left.4\Big/\left(\sqrt{\left(3\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right)\right)\right\},$$

$$\left\{x\to\frac{1}{6}\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}+\right.$$

$$\frac{1}{2}\sqrt{\left(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}+\right.}$$

$$\left.4\Big/\left(\sqrt{\left(3\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right)\right),$$

$$y\to\frac{41}{6}+1\Big/\left(\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right)+$$

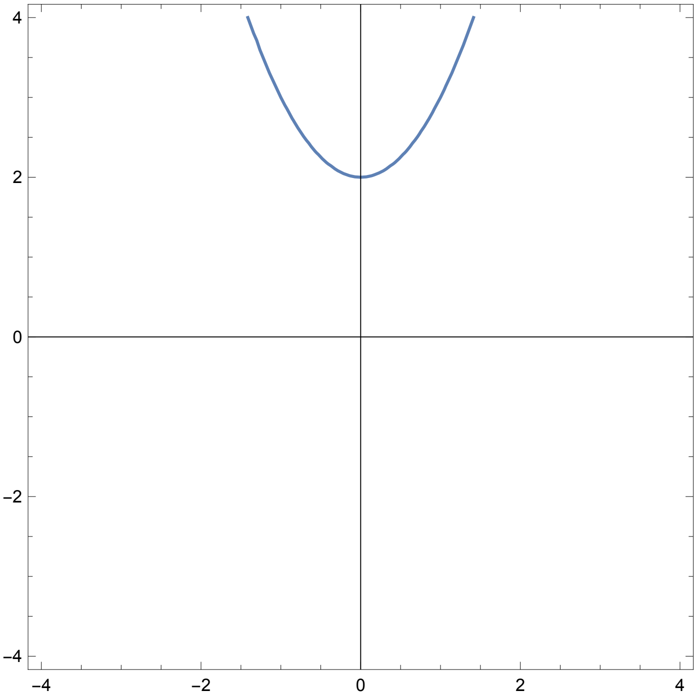
$$\frac{1}{2}\sqrt{\left(\frac{1}{3}\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}$$

$$\sqrt{\left(\frac{140}{27}-\frac{4141}{27\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}-\frac{1}{27}\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}+\right.}$$

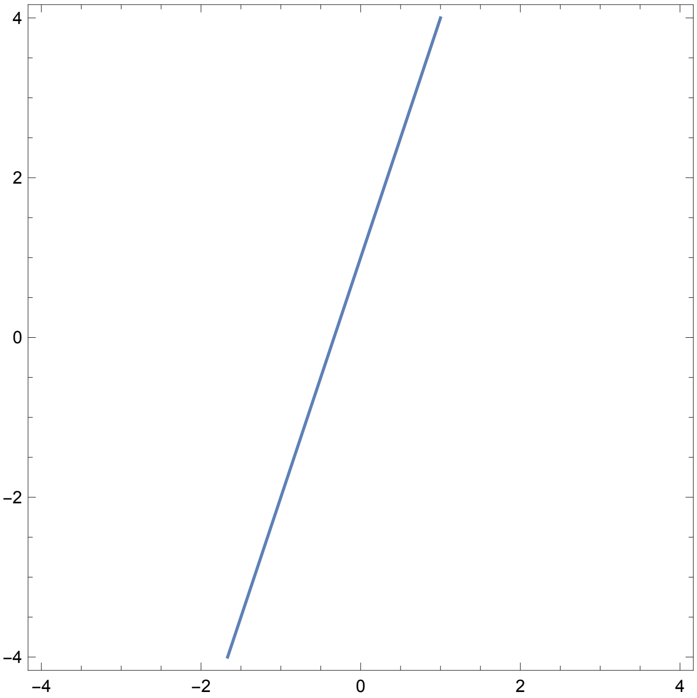
$$\left.\left.4\Big/\left(\sqrt{\left(3\left(70+\frac{4141}{\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}}+\left(263\,791+18\,\mathrm{i}\sqrt{4\,394\,085}\right)^{1/3}\right)\right)}\right)\right)\right\}\right\}$$

The pros of Nsolve are that it produces decimal output and is therefore more simple to understand and digest as well as taking up less physical or virtual space to display. The con is that Nsolve is not an exact expression of the solution but rather an approximation.

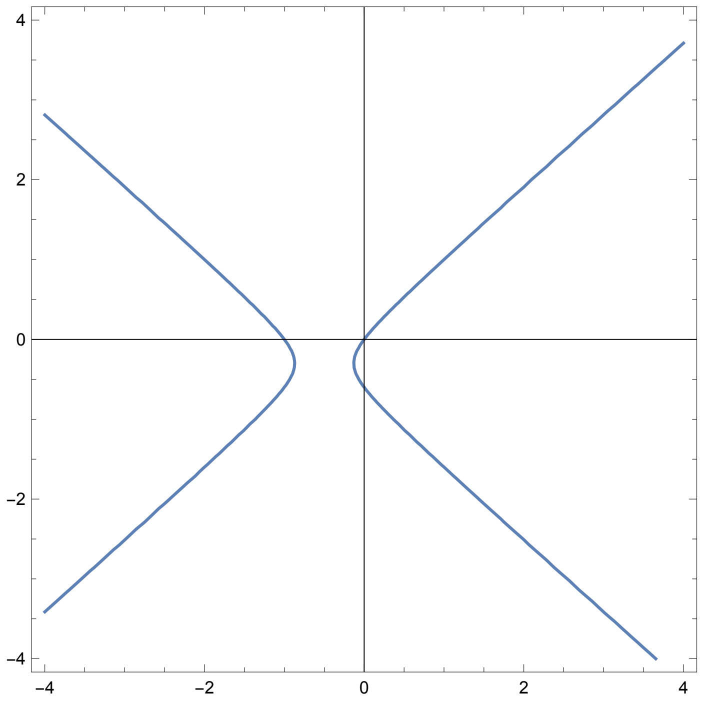
```
ContourPlot[y == x^2 + 2, {x, -4, 4}, {y, -4, 4}, Axes → True]
```



```
ContourPlot[y == 3 x + 1, {x, -4, 4}, {y, -4, 4}, Axes → False]
```

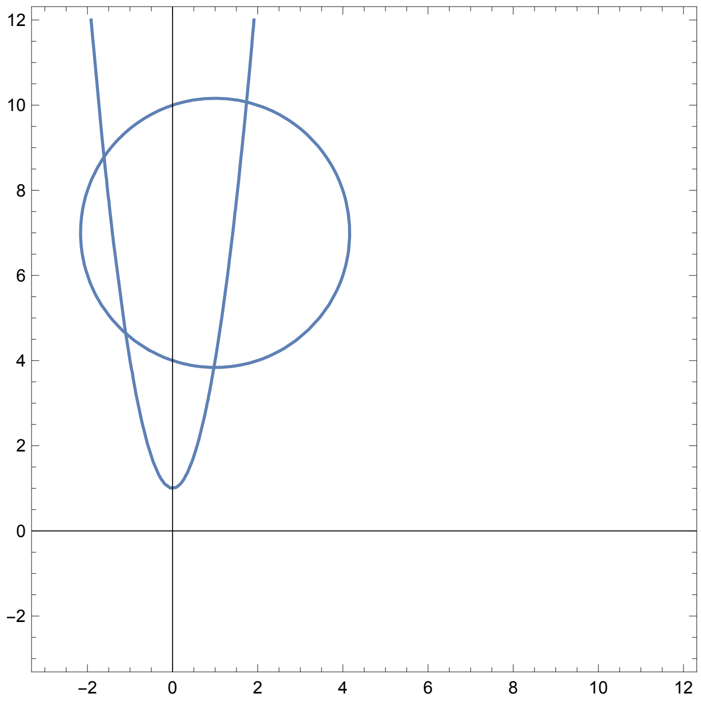


```
ContourPlot[4 x^2 - 5 y^2 + 4 x - 3 y == 0, {x, -4, 4}, {y, -4, 4}, Axes → True]
```



```
In[33]:= Show[ContourPlot[(x - 1)^2 + (y - 7)^2 == 10, {x, -3, 12}, {y, -3, 12}, Axes → True],
          ContourPlot[y == 3 x^2 + 1, {x, -3, 12}, {y, -3, 12}, Axes → True]]
```

Out[33]=

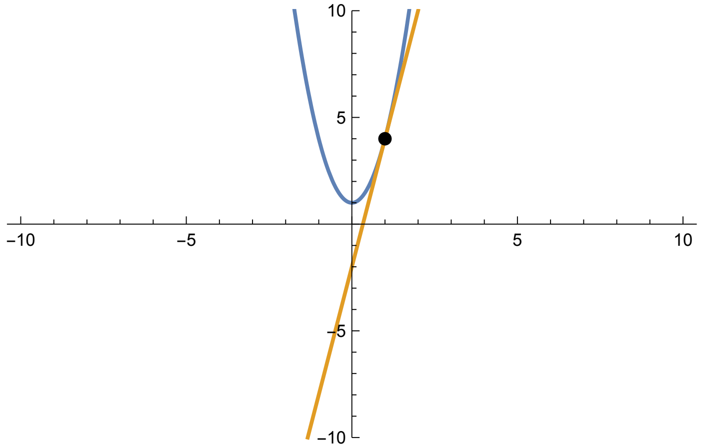


I changed the bounds so that the X-bound and the Y-bound were identical in order to counteract the streching effect seen in the original example.

In[34]:=
```
f[x_] = 3 x^2 + 1
With[{a = 1}, Plot[{f[x], f'[a] (x - a) + f[a]}, {x, -10, 10}, PlotRange → {-10, 10},
  PlotStyle → Thick, Epilog → {PointSize[Large], Point[{a, f[a]}]}]]
```

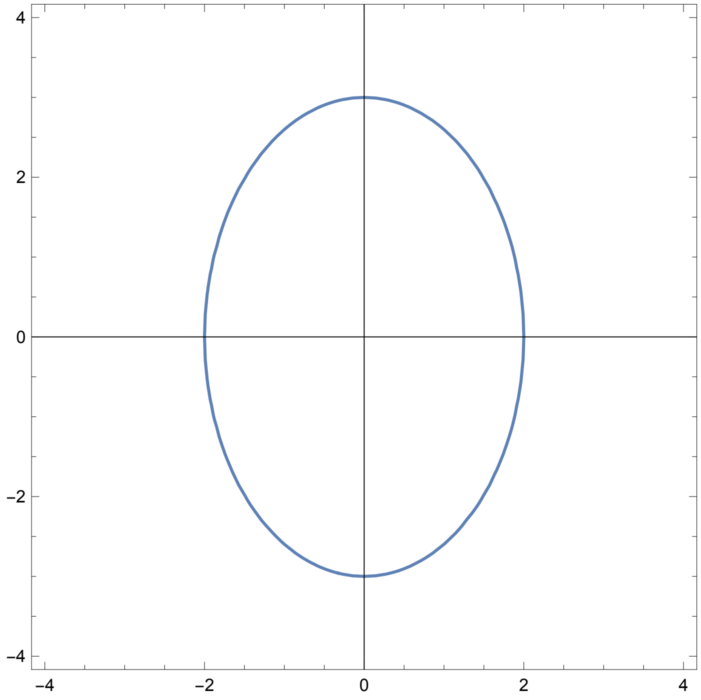
Out[34]= $1 + 3\,x^2$

Out[35]=



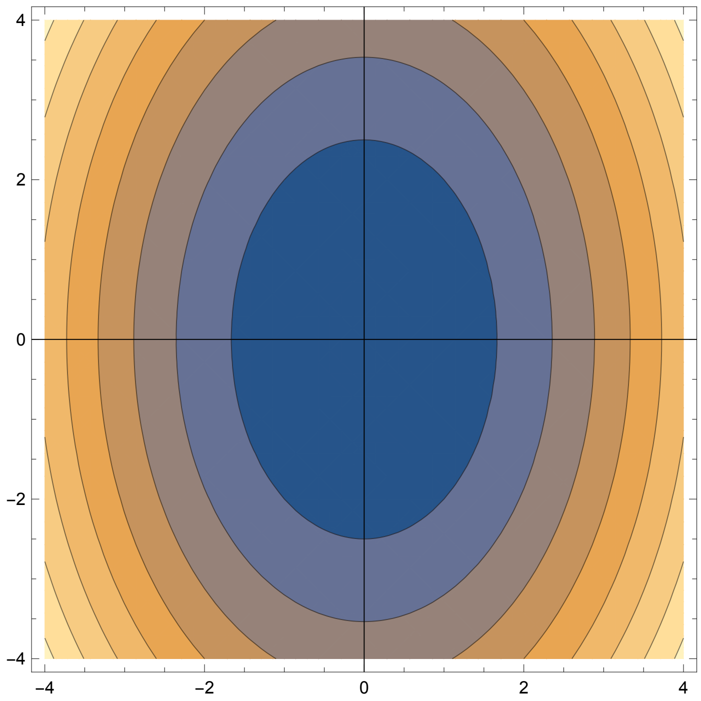
In[36]:=
```
ContourPlot[9 x^2 + 4 y^2 == 36, {x, -4, 4}, {y, -4, 4}, Axes → True]
```

Out[36]=



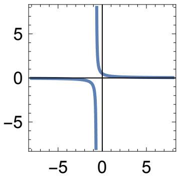
```
In[37]:= ContourPlot[9 x^2 + 4 y^2, {x, -4, 4}, {y, -4, 4}, Axes → True]
```

Out[37]=



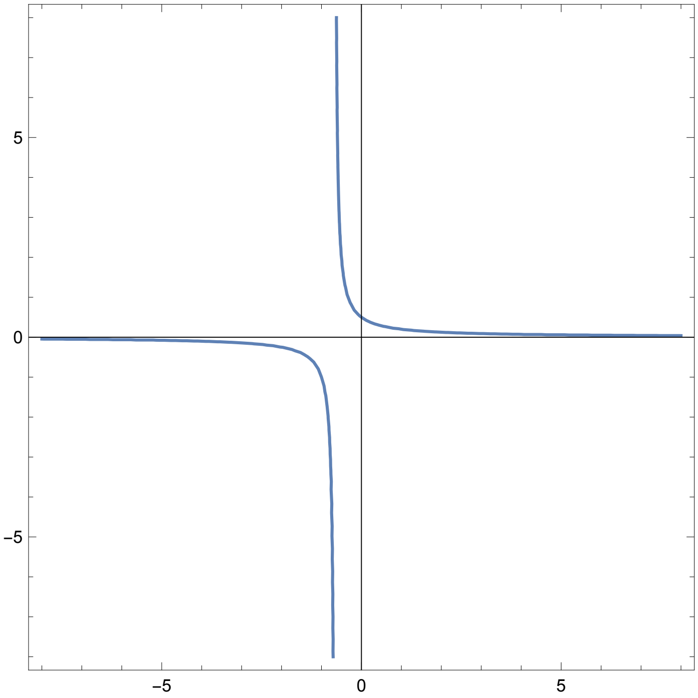
```
In[38]:= levelset[e_, k_] := Module[{k1, set1}, k1 = k;
           set1 = ContourPlot[e == k1, {x, -8, 8}, {y, -8, 8}, Axes → True];
           Return[set1]]
         levelset[3 x y + 2 y, 1]
```

In[40]:=

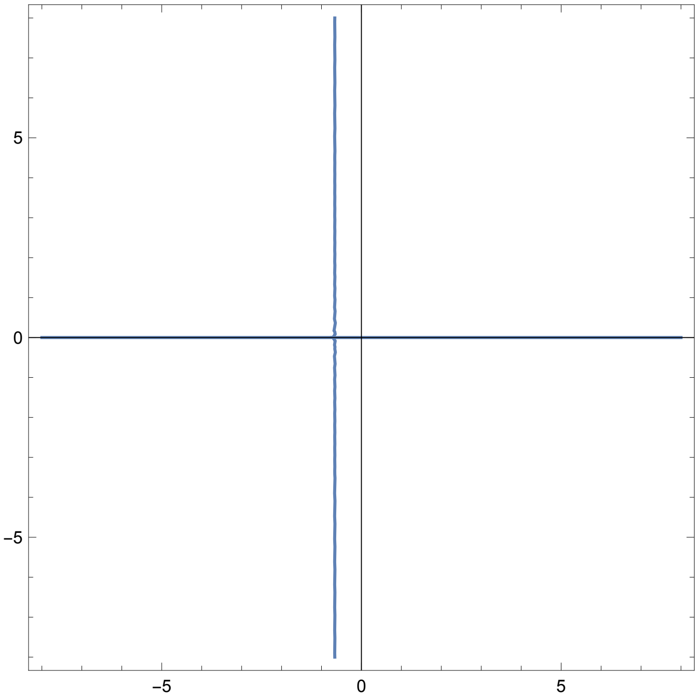


```
levelset[3 x y + 2 y, 0]
```

Out[40]=



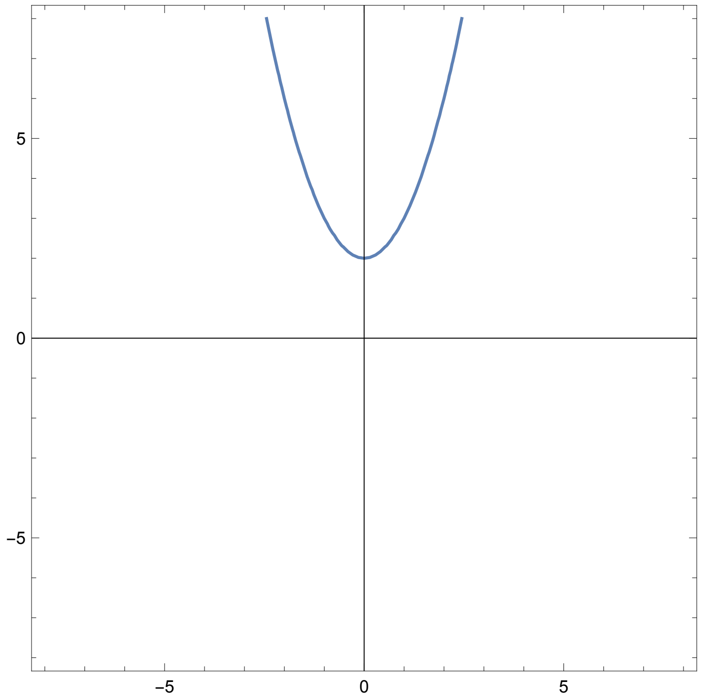
Out[41]=



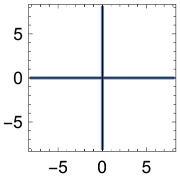
It is not the case that every level set for f(x,y)=3xy+2y is a hyperbola, when you set the constant “k”=0 the level set becomes an orthogonal intersection of lines parallel to the x and y axis rather than a hyperbola.

```
In[54]:= levelset[e_, k_] := Module[{k1, set1}, k1 = k;
           set1 = ContourPlot[e == k1, {x, -8, 8}, {y, -8, 8}, Axes → True];
           Return[set1]]
         levelset[ y - x^2, 2]
```

Out[55]=



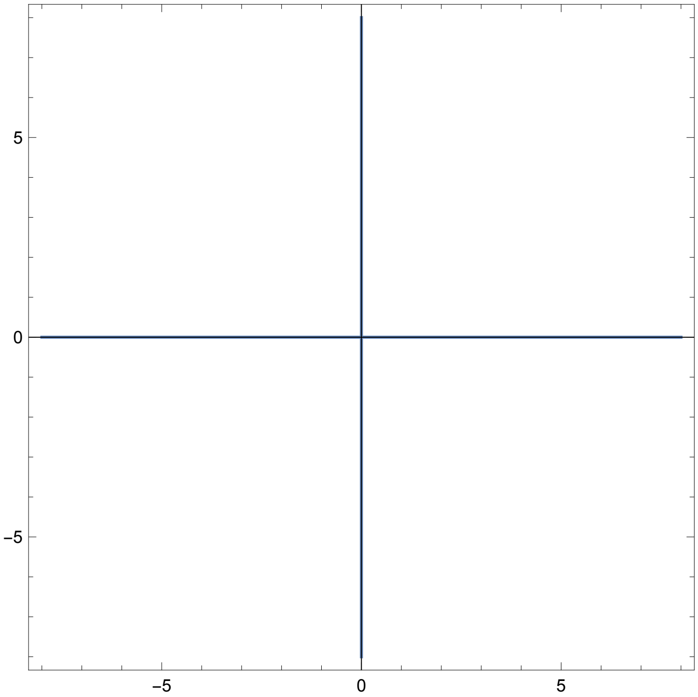
```
In[58]:= levelset[e_, k_] := Module[{k1, set1}, k1 = k;
           set1 = ContourPlot[e == k1, {x, -8, 8}, {y, -8, 8}, Axes → True];
           Return[set1]]
         levelset[x y , 0]
```



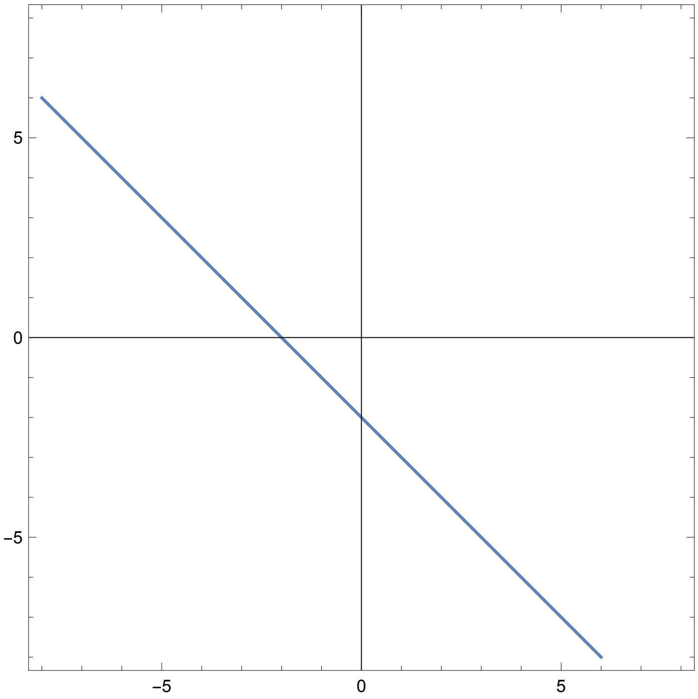
In[102]:=

```
levelset[-.5 x - .5 y , 1]
```

Out[102]=



Out[103]=

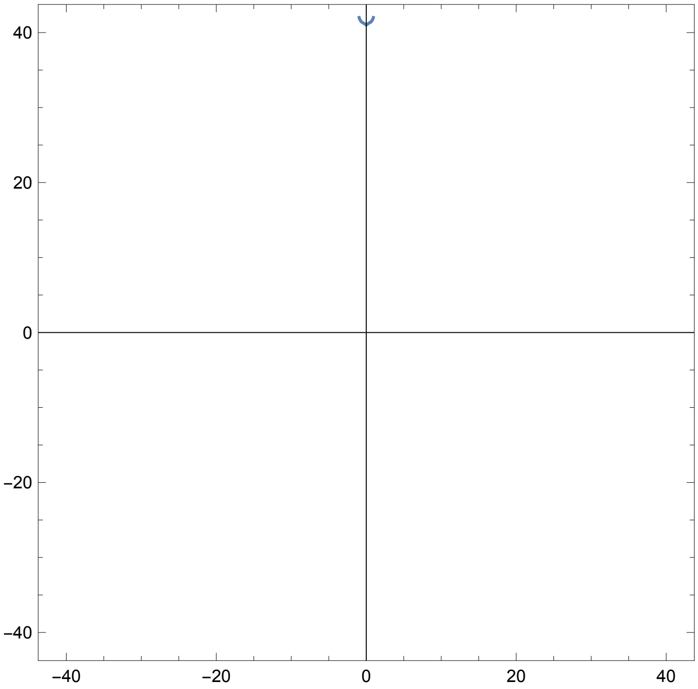


```
In[141]:= levelsetwithbounds[e_, k_, m_] := Module[{k1, set1}, k1 = k;

              set1 = ContourPlot[e == k1, {x, -m, m}, {y, -m, m}, Axes → True]];
          Return[set1]

          levelsetwithbounds[y - x^2, 41, 42]
Out[142]= Return[set1]
```

Out[143]=



```
While::argt : While called with 3 arguments; 1 or 2 arguments are expected. >>

Set::write : Tag Times in 2 set1$25144 While[-m1$25144 ≤ x, y ≤ m1$25144, k1$25144 ≤ abs[40]] is Protected. >>
```

```
In[153]:= log[n_] := Module[{index},
            If[n ≤ 0, Return["5"], Return[Log10[n]]]
           ]
          log[n]

Out[154]= If[n ≤ 0, Return[5], Return[Log10[n]]]

In[148]:= log[100]

Out[148]= 2

In[149]:= log[100 000]

Out[149]= 5

In[150]:= log[13.4]

Out[150]= 1.1271

In[151]:= log[-6]

Out[151]= 5
```

The Module worked as expected, it took the base 10 log of each input, in the case of log[100000] the solution happens by coincidence to be 5 but in the case of -6 the output is 5 because of the If[n<=0,Return["5"]] condition.

```
In[155]:= ilikeevennumbers[n_] := Module[{a}, a = RandomInteger[2];
            If[EvenQ[n], Return[{"Yippee!", "Wonderful", "Fantastic"}[[a + 1]]]]
           ]
          ilikeevennumbers[n]

In[157]:= ilikeevennumbers[2]

Out[157]= Yippee!

In[158]:= ilikeevennumbers[3]

In[165]:= ilikeevennumbers[4]
          ilikeevennumbers[6]
          ilikeevennumbers[8]
          ilikeevennumbers[20]

Out[165]= Fantastic

Out[166]= Wonderful

Out[167]= Yippee!

Out[168]= Fantastic
```

This Module takes an integer input, if it finds it to be an odd number it does not react however if it detects an even number it returns 1 of three phrases starting with "yipee!", each subsequent time it detects an even number a random integer between 0 and 2 is generated and added to 1 corresponding to the phrases positions, because 1 is added every time this process occurs, the process must occur 3 times for any phrase to be reused.

```
In[169]:= sumofsquares[n_] := Module[{index, sum}, index = 1; sum = 0;
             If[Or[n ≤ 0, IntegerQ[n] == False], Return["Please enter a positive integer"]];
             While[index ≤ n, sum = sum + index ^ 2; index = index + 1];
             Return[sum]]
           sumofsquares[n]

Out[170]= Please enter a positive integer

In[171]:= sumofsquares[1]

Out[171]= 1

In[172]:= sumofsquares[2]


Out[172]= 5

In[173]:= sumofsquares[3]

Out[173]= 14

In[174]:= sumofsquares[5]

Out[174]= 55

In[175]:= sumofsquares[100]

Out[175]= 338 350

           positiveint[m_Integer ? (Positive@# &)] := (Print@#;
                Sqrt@Tr@#) &@Range[1, 2 m - 1, 2]

           squarerootofsum[m_] = Module[{index, sum}, index = 0; sum = 0;
             If[Or[m ≤ 0, IntegerQ[m] == False], Return["Please enter a positive integer"]];
             While[index ≤ m,
              sum = Sqrt[Sum[positiveint[m_Integer ? (Positive@# &)] {0, m};] index = index + 1;]

                Return[sum]]]


           squarerootofsum[34]




Out[192]= Return[Please enter a positive integer]

Out[193]= Please enter a positive integer
```