Assignment 19: Euler's Method (7.3) **Name____________________**
Please provide a handwritten response.

1a. To apply Euler's method to the differential equation $y' = \sin y - x^2$ in Exercise 18, Section 7.3, first define $f(x, y) = \sin y - x^2$ by executing

```
f[x_, y_] = Sin[y] - x^2
```

In *Mathematica* functions of two or more variables are handled similarly to functions of one variable; for example, execute **`f[-3, Pi/2]`** to find $f\left(-3, \frac{\pi}{2}\right)$ and record the result below; is it correct?

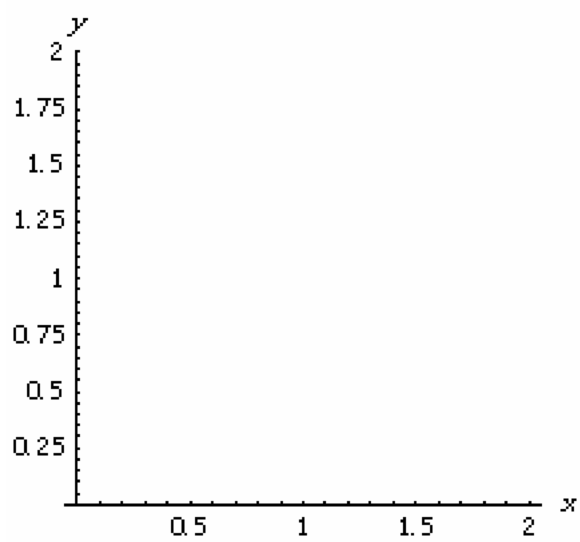
1b. To draw a direction field using we must first load in a package; execute

```
Needs["Graphics`PlotField`"]
```

The following command doesn't fit across one line of this page, but enter and execute it in *Mathematica* as though it did — don't insert any carriage returns of your own.

```
df = PlotVectorField[{1, f[x, y]}, {x, 0, 2}, {y, 0, 2},
  Axes->True, ScaleFunction->(.1&), ScaleFactor->None]
```

Roughly sketch the result on the axes at right. (The **`PlotVectorField`** command draws at the point (x, y) an arrow in the plane whose slope is $\frac{f(x, y)}{1}$, or $f(x, y)$; following the arrows therefore leads to a curve which is a solution to the differential equation. The **`ScaleFunction`** and **`ScaleFactor`** options govern the lengths of the arrows; change the **`.1`** to **`.2`** and see what happens.)



1c. We will store the steps of Euler's method in what *Mathematica* calls a "list"; at each step we add one more ordered pair to our list using the **`AppendTo`** command. For example, execute (using braces, not parentheses!)

```
sample = {{-2, 3}}
```

representing a list consisting of the one ordered pair $(-2, 3)$, and then execute

```
AppendTo[sample, {-3, Pi/2}]
```

What did this do to **`sample`** ?

1d. Execute `{3,-2} + {4, 7}` and record the result below; what do you think happened here? (This "list addition" will come in handy in part **g**.)

1e. At each step of Euler's method we must evaluate $f(x,y)$ at the last ordered pair in our list, in order to compute the next ordered pair. This will take just a bit of fancy stuff: Execute `Last[sample]` and tell below what you think the `Last` command does to a list.

1f. Because *Mathematica* will not do much with `f[Last[sample]]` (try it), we need the `@@` command to "apply" `f` to the two numbers in `Last[sample]` . (The `@` symbol is located above the "2" on your keyboard.) Earlier we calculated $f\left(-3,\frac{\pi}{2}\right)$; now execute `f@@Last[sample]` and record the result below. Is it correct?

1g. Now we can go ahead with the Exercise. Execute `m = {{0, 1}}` to begin our list `m` with the initial condition $y(0)=1$ given in the Exercise, followed by `h = 0.1` to set the step size h . To approximate $y(2)$ will take 20 steps starting from $x_0 = 0$. Using the `Table` command we now can generate our list of Euler steps all at once; execute

```
Table[AppendTo[m, Last[m] + {h, h*f@@Last[m]}], {i, 1, 20}];
```

The semicolon will suppress the output, but then simply execute `m` to see the points generated. What are $y(1)$ and $y(2)$ according to this approximation?

1h. `ListPlot` will plot the points of our list, and the `PlotJoined` and `PlotStyle` options specify whether the graph is connected and the color in which it is drawn. Execute

```
euler1 = ListPlot[m, PlotJoined->True, PlotStyle->Hue[0]]
```

1i. Execute `Clear[m, h]` , followed by `h = 0.05` . Re–execute the other commands of part **g** <u>after</u> you change `20` to `40` , and tell below the values of $y(1)$ and $y(2)$ according to this approximation.

1j. Now execute

```
euler2 = ListPlot[m, PlotJoined->True, PlotStyle->Hue[.7]]
```

followed by `Show[df, euler1, euler2]` to combine the graphs, and sketch the result on your direction field in part **b**, labeling both curves clearly.