Experiment 3

In [1]:
```
from matplotlib import pylab
```

In [2]:
```
print(pylab.__version__)
```

```
1.23.3
```

In [3]:
```
import numpy as np
```

In [4]:
```
x = np.linspace(0,10,25)
```

In [5]:
```
y = x * x + 2
```

In [6]:
```
print(x)
```

```
[ 0.          0.41666667  0.83333333  1.25        1.66666667  2.08333333
  2.5         2.91666667  3.33333333  3.75        4.16666667  4.58333333
  5.          5.41666667  5.83333333  6.25        6.66666667  7.08333333
  7.5         7.91666667  8.33333333  8.75        9.16666667  9.58333333
 10.        ]
```

In [7]:
```
print(y)
```

```
[  2.           2.17361111   2.69444444   3.5625       4.77777778
   6.34027778   8.25        10.50694444  13.11111111  16.0625
  19.36111111  23.00694444  27.          31.34027778  36.02777778
  41.0625      46.44444444  52.17361111  58.25        64.67361111
  71.44444444  78.5625      86.02777778  93.84027778 102.        ]
```

In [8]:
```
print(np.array([x,y]).reshape(25,2))
```

```
[[  0.           0.41666667]
 [  0.83333333   1.25      ]
 [  1.66666667   2.08333333]
 [  2.5          2.91666667]
 [  3.33333333   3.75      ]
 [  4.16666667   4.58333333]
 [  5.           5.41666667]
 [  5.83333333   6.25      ]
 [  6.66666667   7.08333333]
 [  7.5          7.91666667]
 [  8.33333333   8.75      ]
 [  9.16666667   9.58333333]
 [ 10.           2.        ]
 [  2.17361111   2.69444444]
 [  3.5625       4.77777778]
 [  6.34027778   8.25      ]
 [ 10.50694444  13.11111111]
 [ 16.0625      19.36111111]
 [ 23.00694444  27.        ]
 [ 31.34027778  36.02777778]
 [ 41.0625      46.44444444]
 [ 52.17361111  58.25      ]
 [ 64.67361111  71.44444444]
 [ 78.5625      86.02777778]
 [ 93.84027778 102.        ]]
```

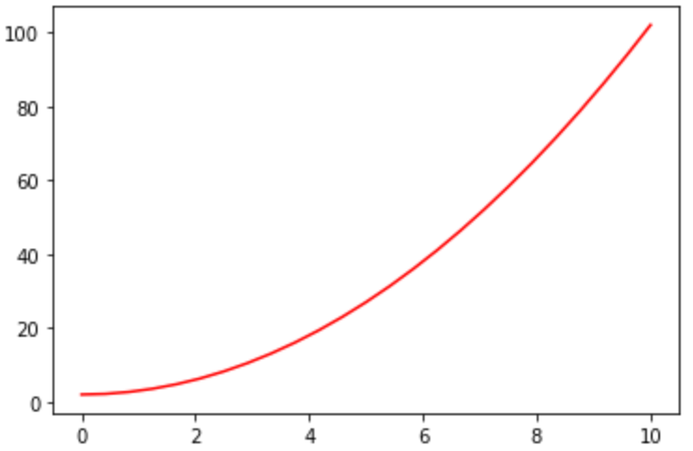
In [9]:
```
print(np.array([x,y]).reshape(2,25))
```

```
[[  0.           0.41666667   0.83333333   1.25         1.66666667
```

```
    2.08333333   2.5          2.91666667   3.33333333   3.75
    4.16666667   4.58333333   5.           5.41666667   5.83333333
    6.25         6.66666667   7.08333333   7.5          7.91666667
    8.33333333   8.75         9.16666667   9.58333333  10.        ]
[   2.           2.17361111   2.69444444   3.5625       4.77777778
    6.34027778   8.25        10.50694444  13.11111111  16.0625
   19.36111111  23.00694444  27.          31.34027778  36.02777778
   41.0625      46.44444444  52.17361111  58.25        64.67361111
   71.44444444  78.5625      86.02777778  93.84027778 102.        ]]
```

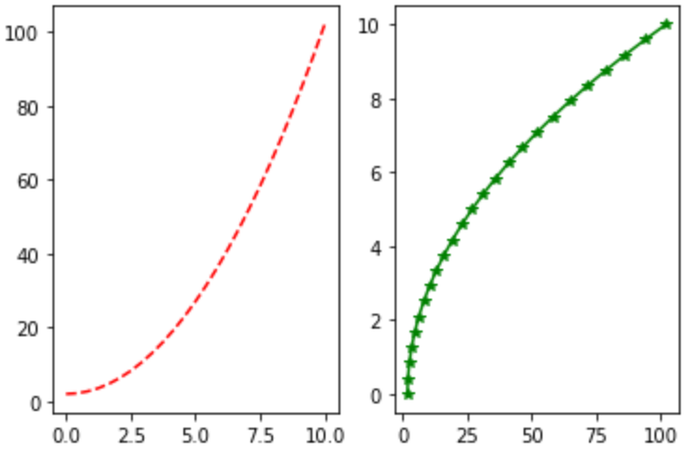
In [10]:
```
pylab.plot(x,y,'r')
#Creates a plain canvas and draws the graph
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x7ff668c005e0>]
```



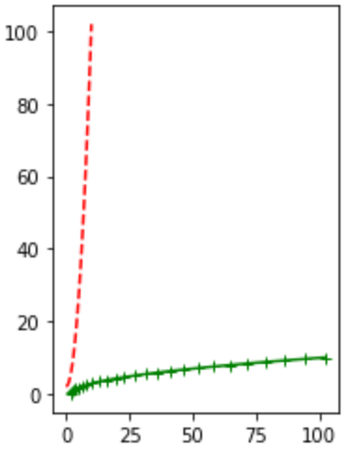
In [11]:
```
pylab.subplot(1,2,1) #row, columns, index of current graph
pylab.plot(x,y, 'r--') #Third parameter denotes the color and the line style
pylab.subplot(1,2,2)
pylab.plot(y,x, 'g*-')
```

Out[11]:
```
[<matplotlib.lines.Line2D at 0x7ff6380af9d0>]
```



In [12]:
```
pylab.subplot(1,2,1) #row, columns, index of current graph
pylab.plot(x,y, 'r--') #Third parameter denotes the color and the line style
pylab.subplot(1,2,1)
pylab.plot(y,x, 'g+-')
```

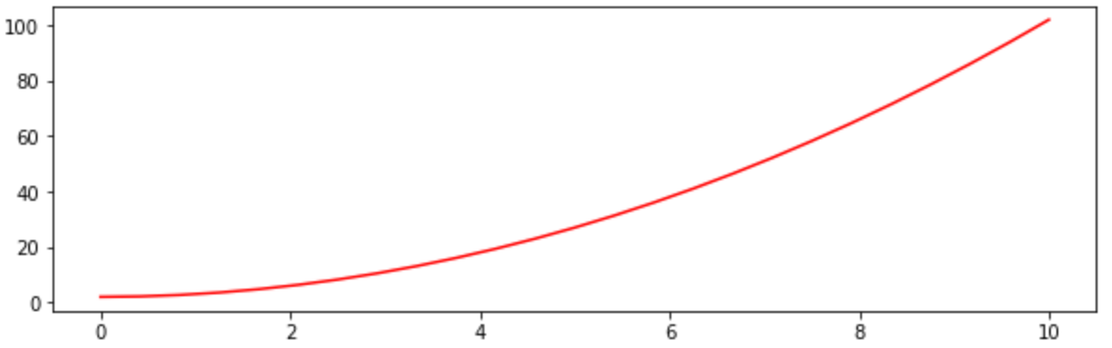
Out[12]:
```
[<matplotlib.lines.Line2D at 0x7ff67a79a760>]
```



In [13]:
```
from matplotlib import pyplot as plt
```

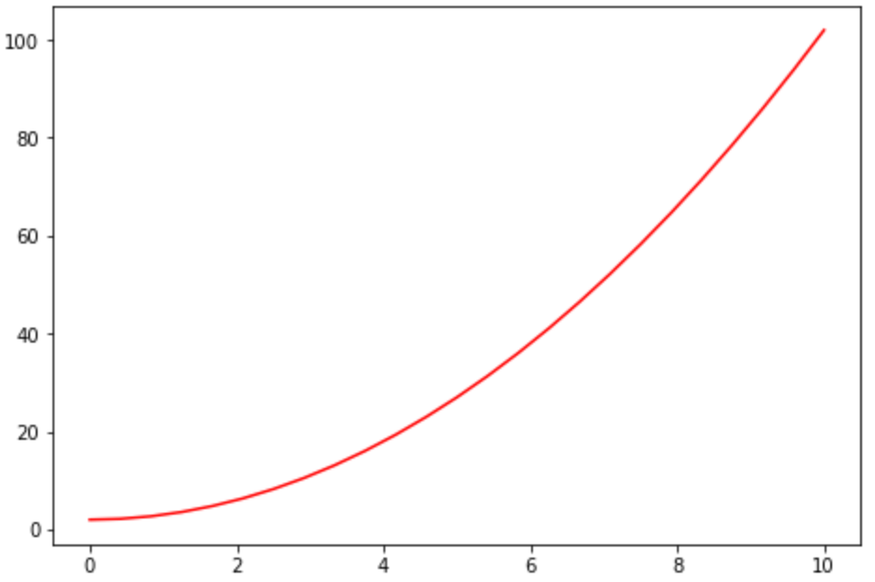
In [14]:
```
fig = plt.figure(figsize = (10,3)) #Change the y axis size and explore
plt.plot(x,y,'r')
```

Out[14]:
```
[<matplotlib.lines.Line2D at 0x7ff67a6a1520>]
```



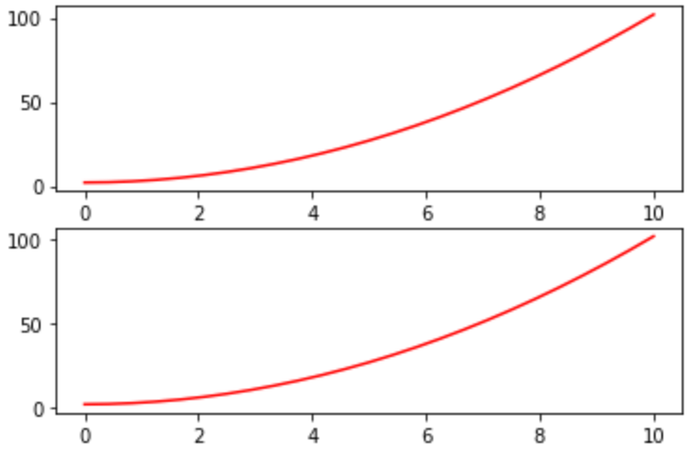
In [15]:
```
fig = plt.figure()
axis = fig.add_axes([1,1,1,1]) #Controls the left, bottom, width and height␣,(0-1)
axis.plot(x,y,'r') #change the axes proportions and explore
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x7ff6480967f0>]
```



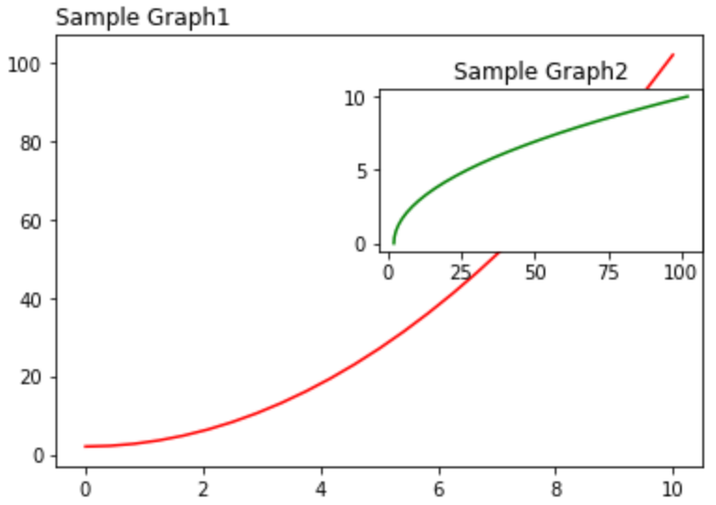
In [16]:
```
fig, axes = plt.subplots(nrows=2, ncols=1) #Change the values and explore
for ax in axes:
```

```
    ax.plot(x,y,'r')
```



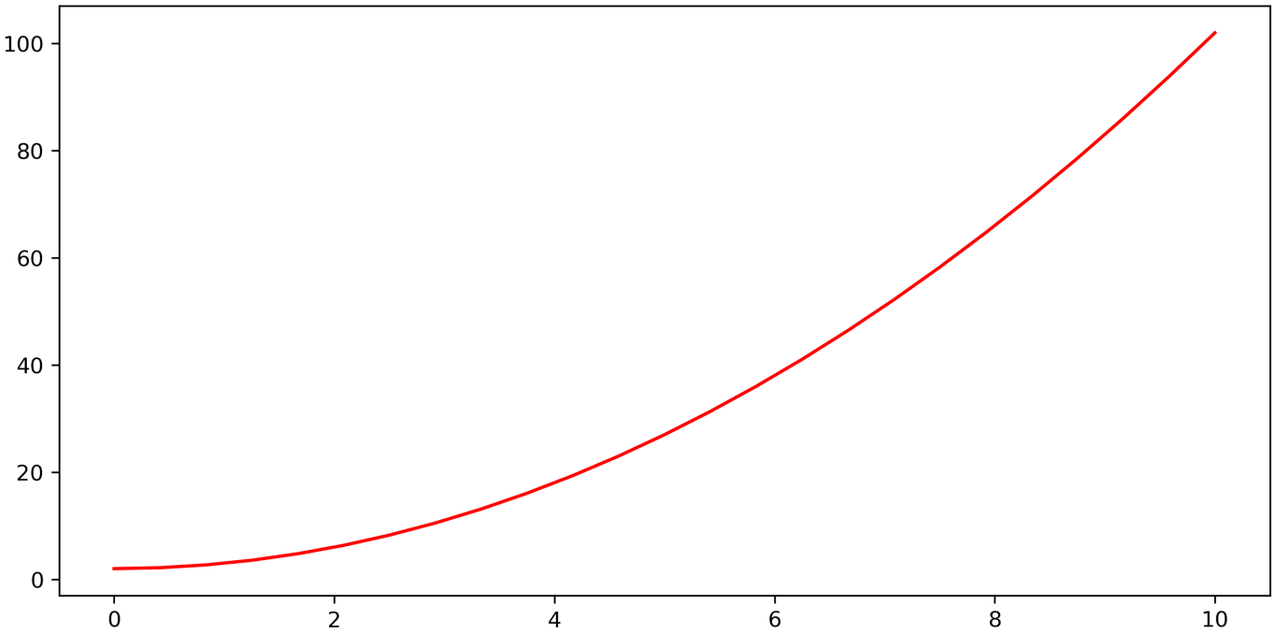
In [17]:
```
fig = plt.figure()
axes1 = fig.add_axes([0.1,0.1,0.8,0.8]) #bigger Canvas
axes1 = plt.plot(x,y,'r')
plt.title("Sample Graph1", loc = 'left')
axes2 = fig.add_axes([0.5,0.5,0.4,0.3]) #Smaller Canvas
axes2 = plt.plot(y,x, 'g') #Change the order of statements and explore
plt.title("Sample Graph2")
```

Out[17]:
```
Text(0.5, 1.0, 'Sample Graph2')
```



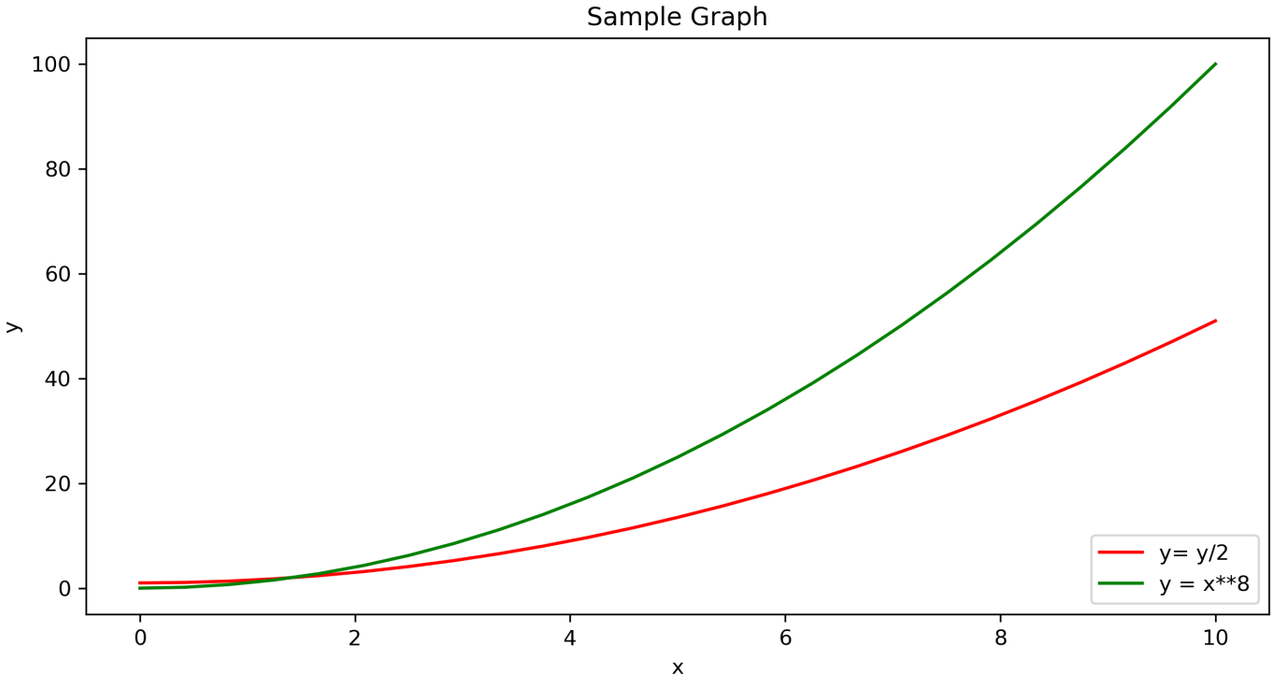
In [18]:
```
fig = plt.figure(figsize = (10,5), dpi = 300)
plt.plot(x,y,'r') #increasing the dpi will take time to process the graph
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x7ff668f6ba00>]
```



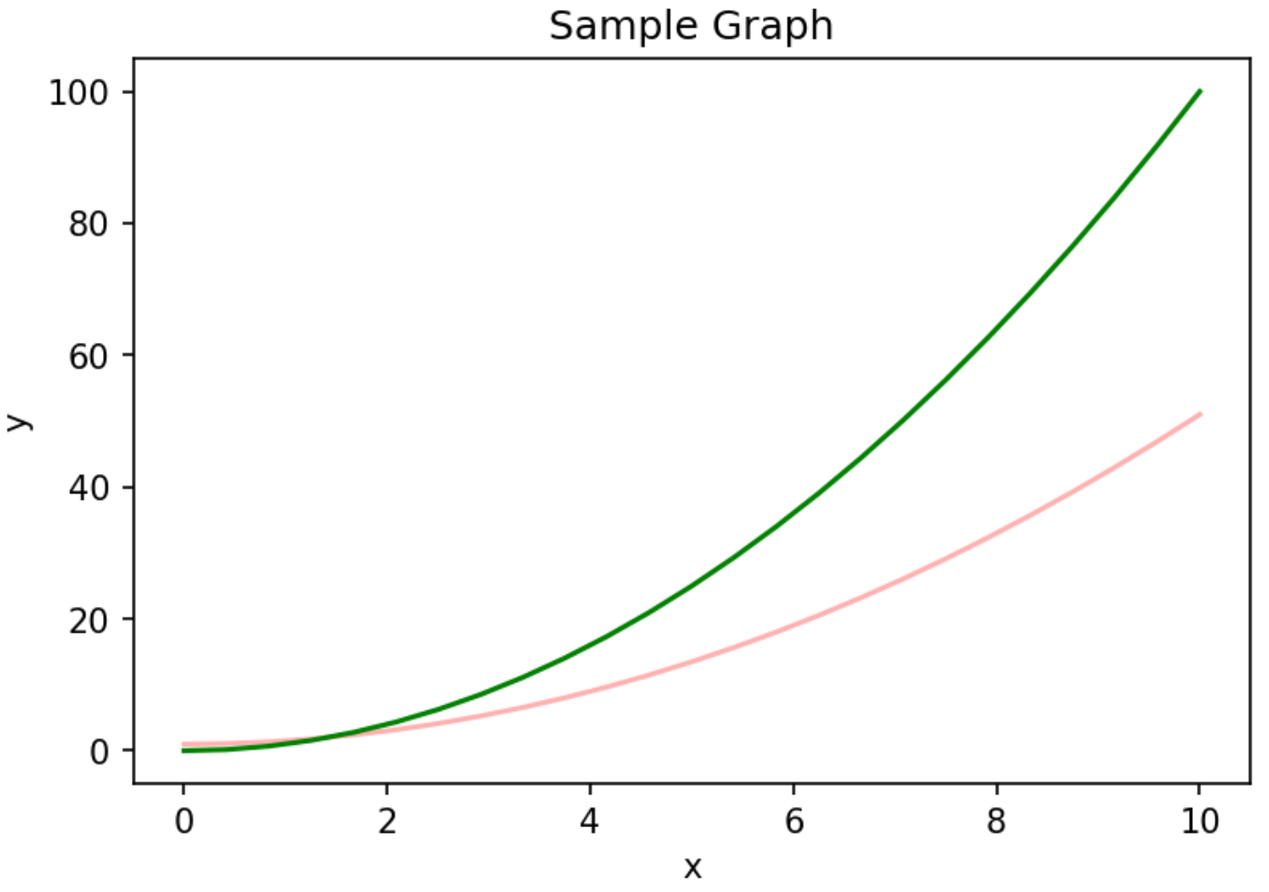
In [19]:
```
 fig, axes = plt.subplots(figsize = (10,5), dpi=300)
axes.plot(x,y/2, 'r')
axes.plot(x, x**2, 'g')
axes.set_title("Sample Graph")
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.legend(['y= y/2', 'y = x**8'], loc=4)
```

Out[19]:
```
<matplotlib.legend.Legend at 0x7ff6480cb400>
```



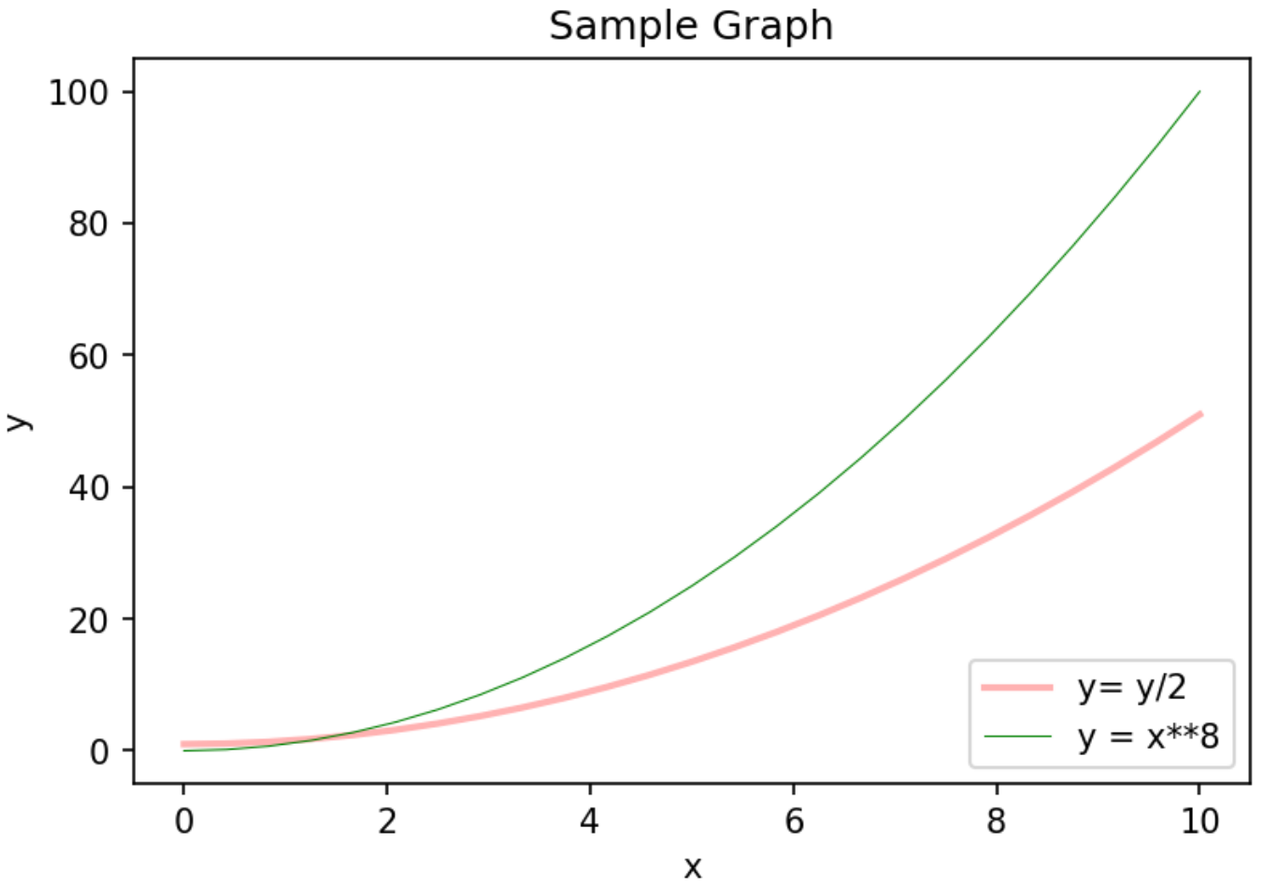
In [20]:
```
fig, axes = plt.subplots(dpi=150)
axes.plot(x,y/2, 'r', alpha=0.3)
axes.plot(x, x**2, 'g')
axes.set_title("Sample Graph")
axes.set_xlabel('x')
axes.set_ylabel('y')
```

Out[20]:
```
Text(0, 0.5, 'y')
```



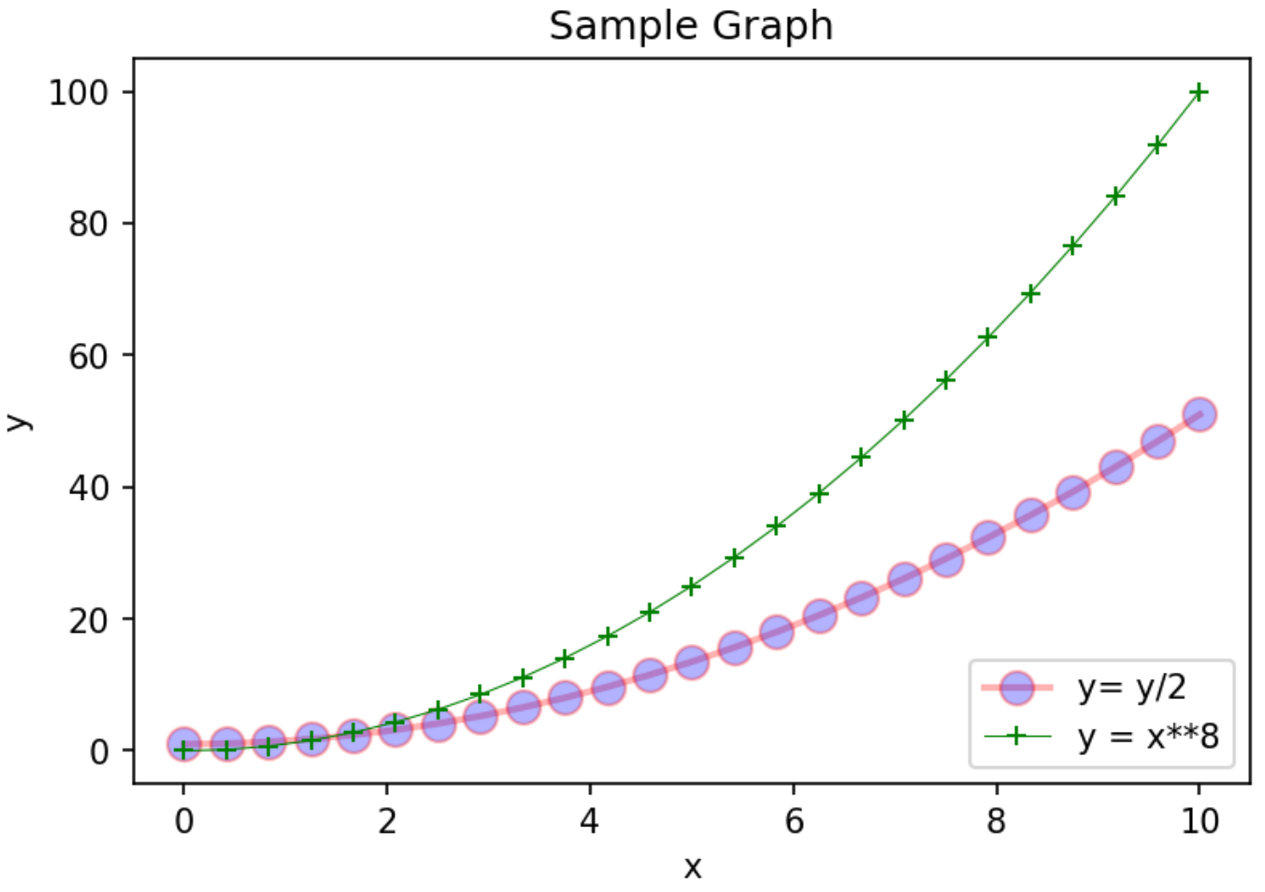
In [21]:
```
fig, axes = plt.subplots(dpi=150)
axes.plot(x,y/2, 'r', alpha=0.3, lw = 2)
axes.plot(x, x**2, 'g', linewidth=0.5)
axes.set_title("Sample Graph")
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.legend(['y= y/2', 'y = x**8'], loc=4)
```

Out[21]:
```
<matplotlib.legend.Legend at 0x7ff6886546d0>
```

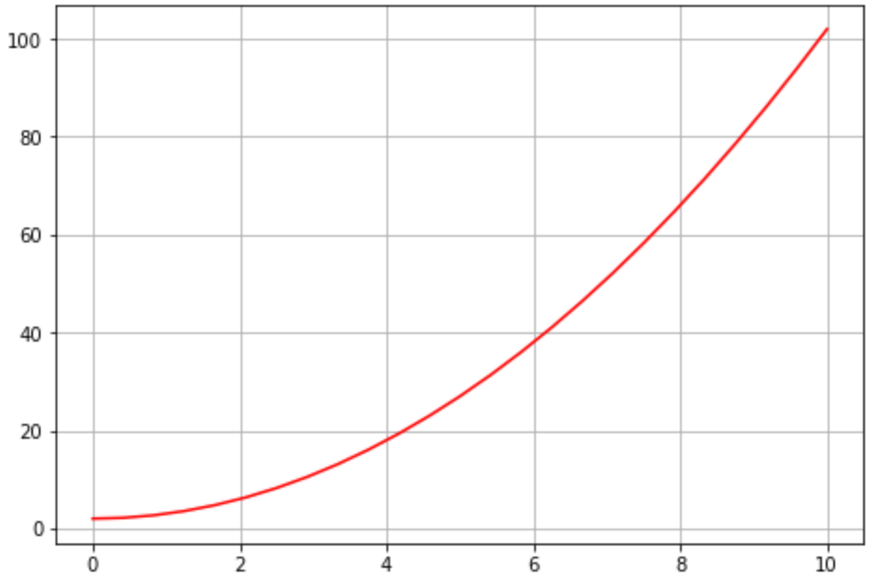


In [22]:
```
fig, axes = plt.subplots(dpi=150)
axes.plot(x,y/2, 'r', alpha=0.3, lw = 2, marker ='o', markersize =10,markerfacecolor ='b
axes.plot(x, x**2, 'g', linewidth=0.5, marker = '+')
axes.set_title("Sample Graph")
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.legend(['y= y/2', 'y = x**8'], loc=4)
```

Out[22]:
```
<matplotlib.legend.Legend at 0x7ff668f2fd60>
```

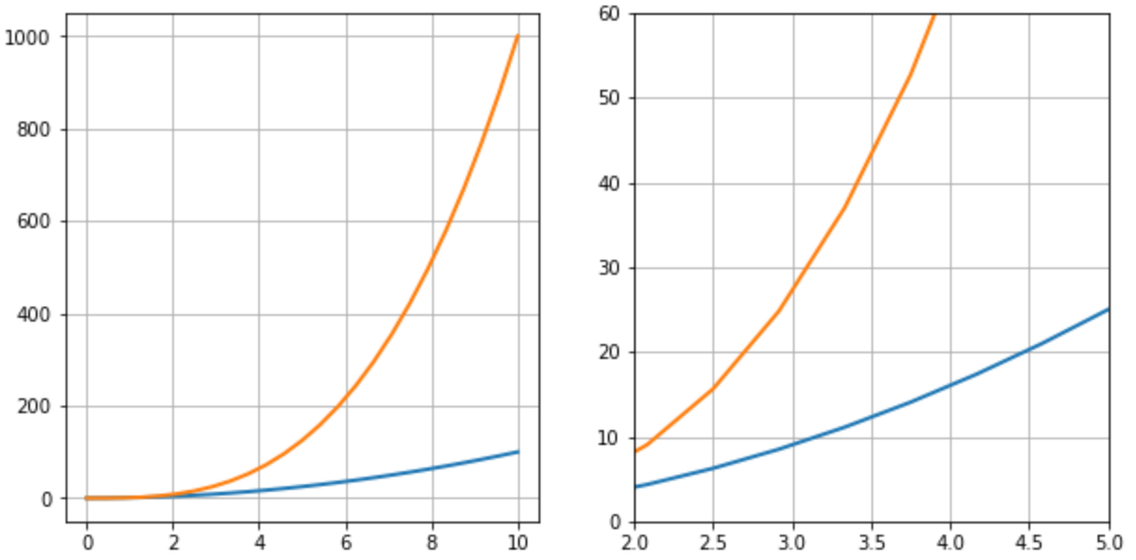


In [23]:
```
fig = plt.figure()
axis = fig.add_axes([1,1,1,1]) #Controls the left, bottom, width and height␣,(0-1)
axis.plot(x,y,'r')
axis.grid(True)
```



In [24]:
```
fig, axes = plt.subplots(1,2, figsize = (10,5))
axes[0].plot(x,x**2, x, x**3, lw=2)
axes[0].grid(True)
axes[1].plot(x,x**2, x, x**3, lw=2)
axes[1].grid(True)
axes[1].set_xlim([2,5])
axes[1].set_ylim([0,60])
```

Out[24]: (0.0, 60.0)



In [25]:
```
import seaborn as sns
```

In [26]:
```
conda install seaborn
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Retrieving notices: ...working... done

Note: you may need to restart the kernel to use updated packages.
```

In [27]:
```
sns.color_palette()
```

Out[27]:

In [ ]: