In [1]:
```
from sympy import *
x = Symbol('x')
d = diff(x**2, x,1)
d
```

Out[1]: $\displaystyle 2 x$

In [6]:
```
d = diff(sin(x),x,1)
d
```

Out[6]: $\displaystyle \cos{\left(x \right)}$

In [4]:
```
d = diff(exp(x),x,1)
d
```

Out[4]: $\displaystyle e^{x}$

In [9]:
```
y = Symbol('y')
d = diff(2*x**2 + 2*y**2, x,1)
d
```

Out[9]: $\displaystyle 4 x$

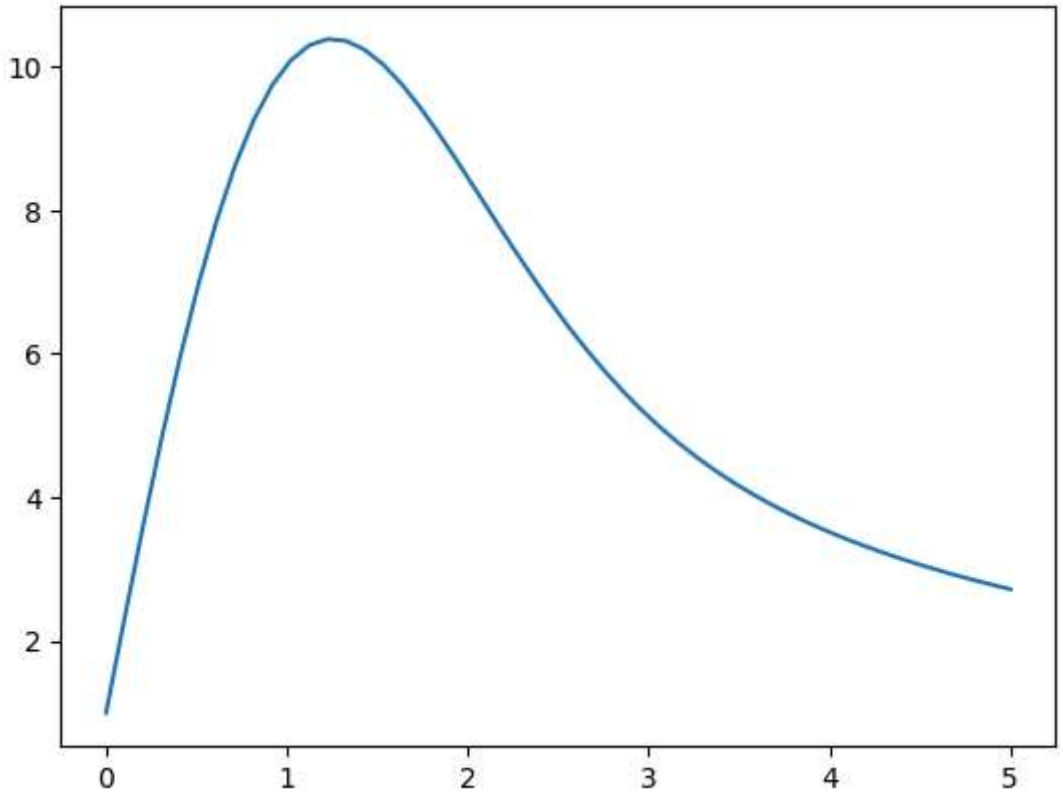
In [10]:
```
d = diff(sqrt(x**3)+4*x**2, x, 1)
d
```

Out[10]: $\displaystyle 8 x + \frac{3 \sqrt{x^{3}}}{2 x}$

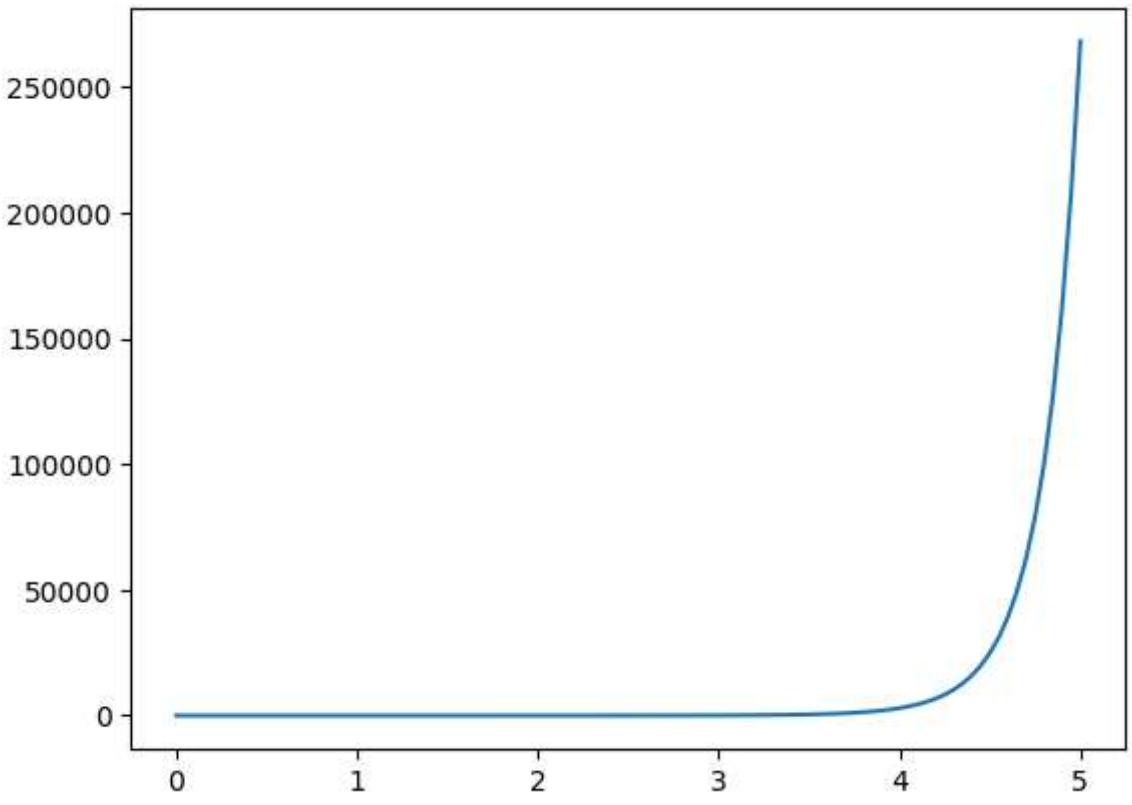
In [11]:
```
d = diff((x**2)/ 8 - log(x), x,1)
d
```

Out[11]: $\displaystyle \frac{x}{4} - \frac{1}{x}$

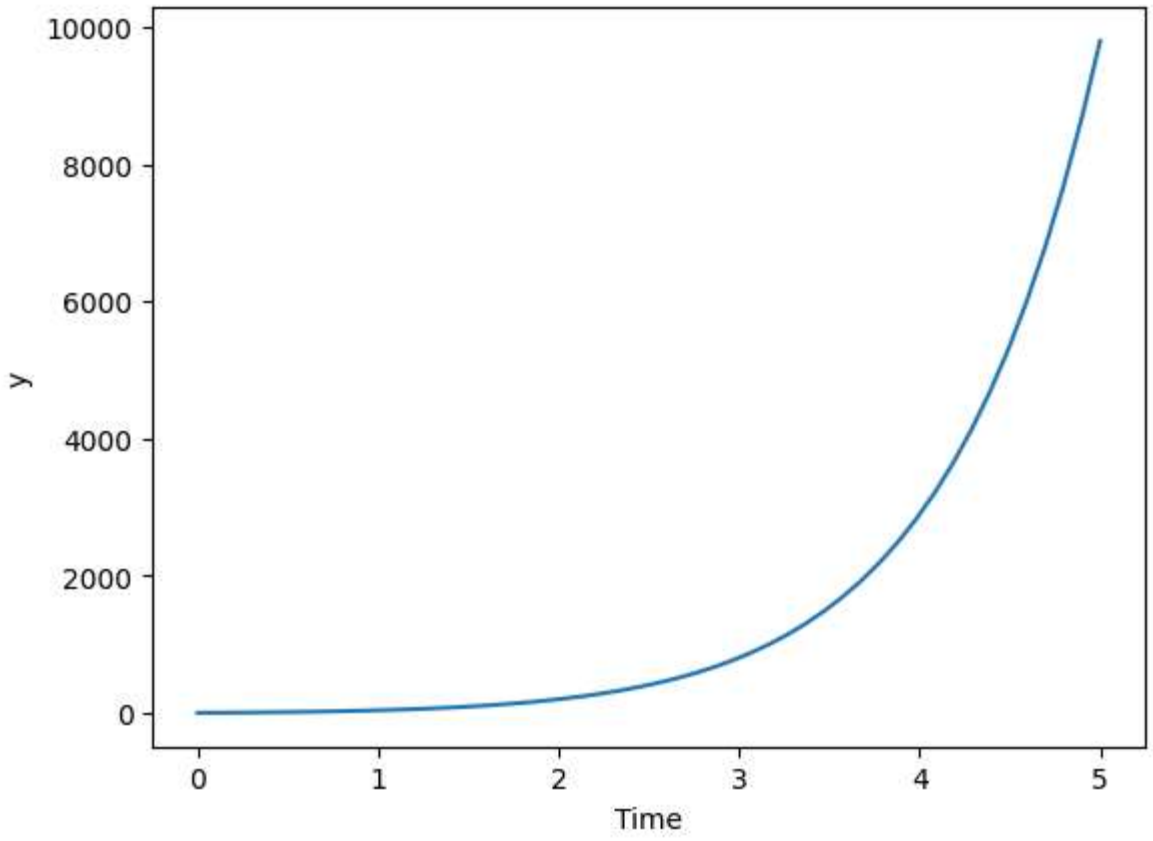
In [2]:
```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

y=1

t = np.linspace(0,5)

def returns_dydt(y, t):
    dydt = -y * t + 13
    return dydt

y = odeint(returns_dydt, y, t)

plt.plot(t,y)
plt.show()
```



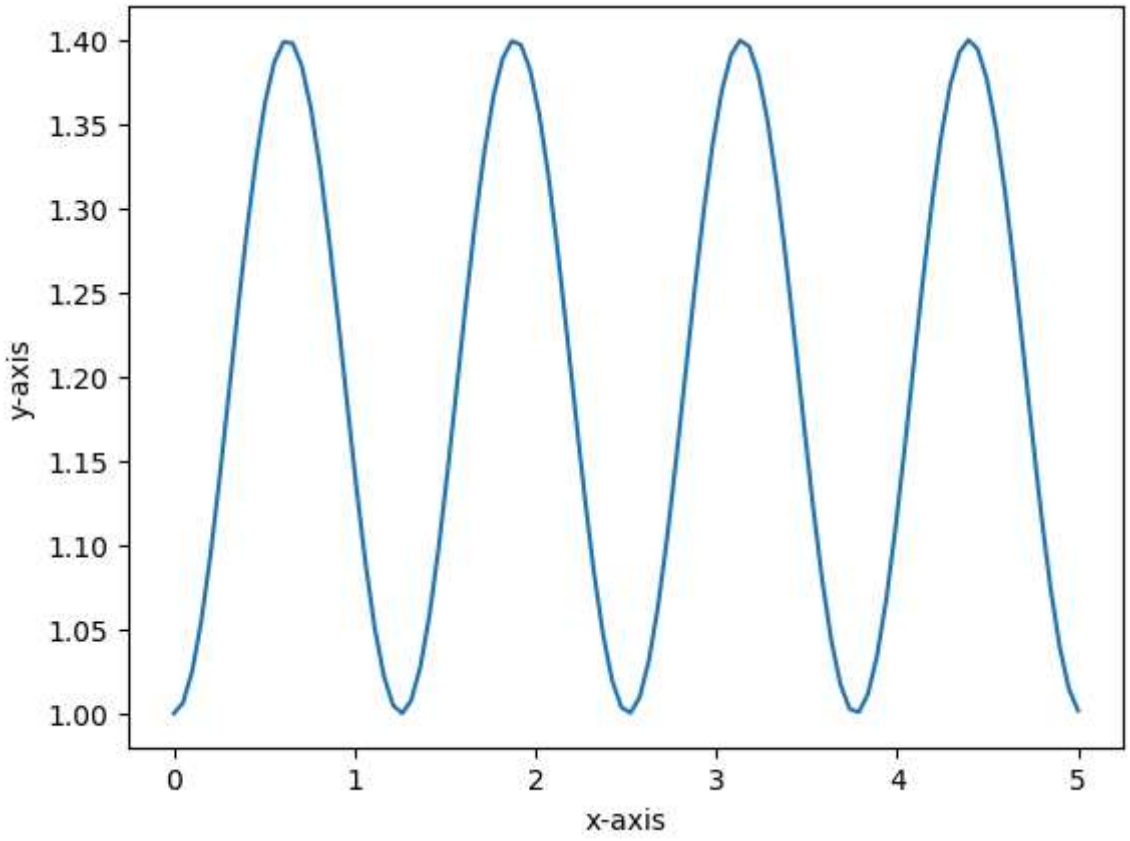
In [3]:
```
def dy_dx(y,x):
    return x*y

y = 1.0
x = np.linspace(0,5,100)
y = odeint(dy_dx, y,x)
plt.plot(x,y)
plt.show()
```



In [3]:
```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

y0=1
#values of time
t=np.linspace(0,5)
def returns_dydt(y,t):
    dydt=13*np.exp(t)+y
    return dydt

y=odeint(returns_dydt,y0,t)

plt.plot(t,y)
plt.xlabel("Time")
plt.ylabel("y")
plt.show()
```



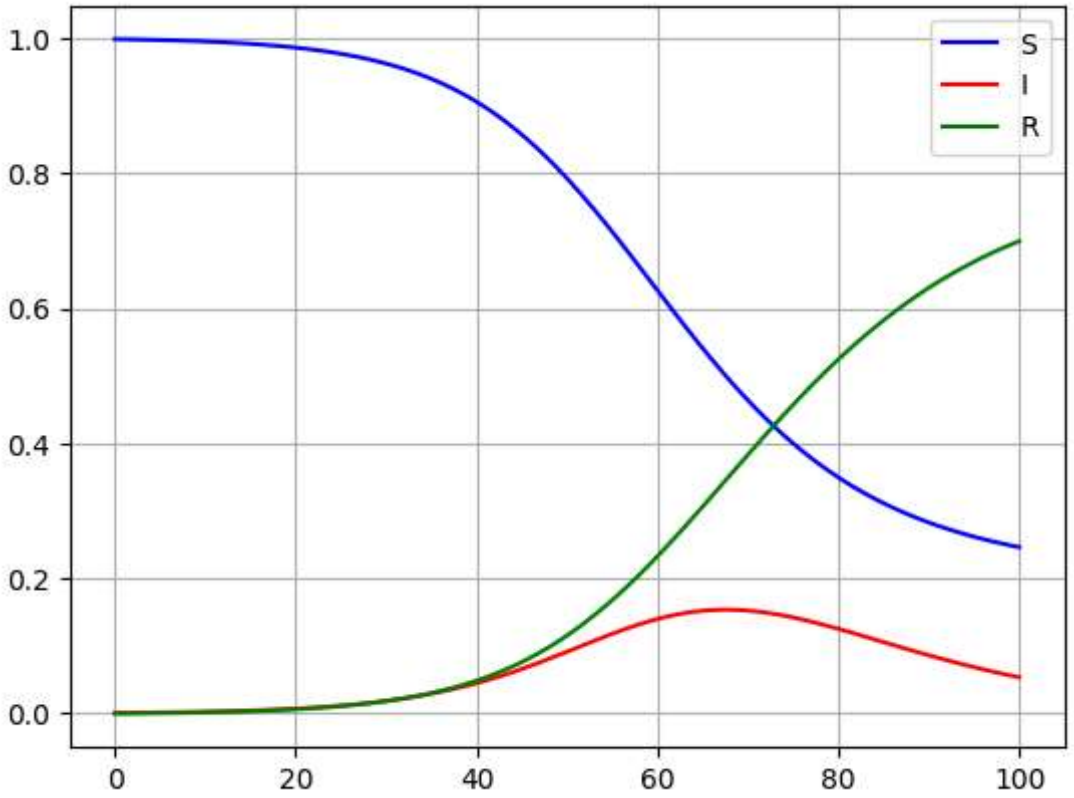
In [4]:
```
y0=1
x=np.linspace(0,5,100)
def dy_dx(y,x):
    return np.sin(5*x)
y=odeint(dy_dx,y0,x) #odeint -inbuilt function
#plot results
plt.plot(x,y)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```



In [8]:

```
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt


beta = .2
gamma = .1
N = 1000

I0 = 1
R0 = 0
S0 = N - I0 - R0

t = np.linspace(0,100,100)

def df(y,t,n, beta,gamma):
    S,I,R = y
    dsdt = -beta*S*I/N
    dIdt = beta*S*I/N - gamma*I
    dRdt = gamma*I
    return dsdt,dIdt,dRdt

y0 = S0,I0,R0
ret = odeint(df,y0,t,args=(N,beta,gamma))
S = ret[:,0]
I = ret[:,1]
R = ret[:,2]

plt.plot(t,S/N,color='blue',label='S')
plt.plot(t,I/N,color='red',label='I')
plt.plot(t,R/N,color='green',label='R')

plt.legend()
```

```
plt.grid()
plt.show()
```



In [ ]: