Lab Assignment 04

1. A culture initially has $P0$ number of bacteria. At t = 1 h the number of bacteria is measured to be $5/2P0$. If the rate of growth is proportional to the number of bacteria $P(t)$ present at time t, determine the time necessary for the number of bacteria to triple.

In [8]:

```
import numpy as np
import math
from numpy import *
from sympy import *
from pylab import *
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from sympy.interactive import printing
printing.init_printing(use_latex=True)
```

In [3]:

```
t,k = Symbol('t'), Symbol('k')
P = Function('P')(t)
diffeq = Eq(P.diff(t)-k*P,0)
sol = dsolve(diffeq)
from sympy.interactive import printing
printing.init_printing(use_latex=True)
P0 = Symbol('P0')
constants = solve([sol.subs([(P,P0),(t,t)]),sol.subs([(P,5/2*P0),(t,t+1)])])
sol = sol.subs(k, constants[1][k])
sol
```

Out[3]:

$$P(t) = C_1 e^{0.916290731874155t}$$

In [4]:

```
c1 = solve([sol.subs([(P,P0),(t,0)]),sol.subs([(P,5/2*P0),(t,1)])])
c1
```

Out[4]:

$$\{C_1 : P_0\}$$

In [6]:

```
sol = sol.subs(c1)
sol
```

Out[6]:

$$P(t) = P_0 e^{0.916290731874155t}$$

In [7]:

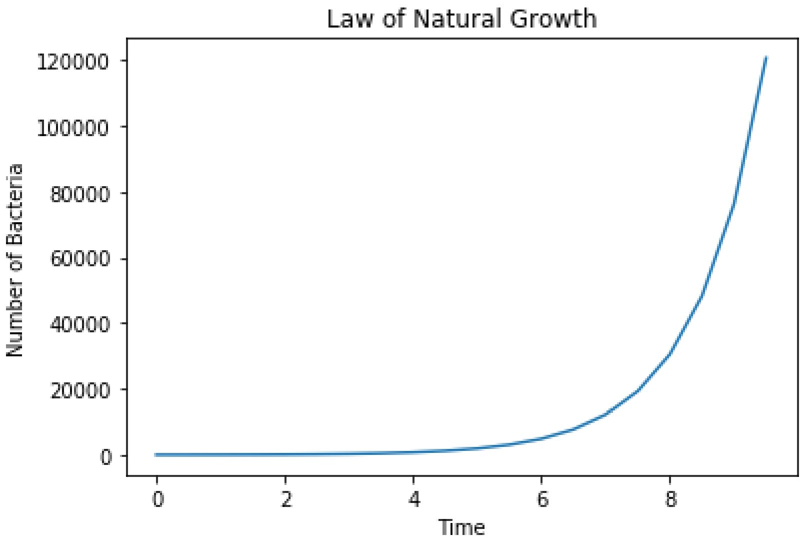
```
solve([sol.subs(P,3*P0)],t)
```

Out[7]:

$\{t : 1.19897784671579\}$

In [10]:

```
t = arange(0,10,0.5)
P0 = 20
y=20*exp(0.916290731874155*t)
plt.plot(t,y)
plt.xlabel('Time')
plt.ylabel('Number of Bacteria')
plt.title('Law of Natural Growth')
plt.show()
```



2. Expain Logistic growth with two examples.

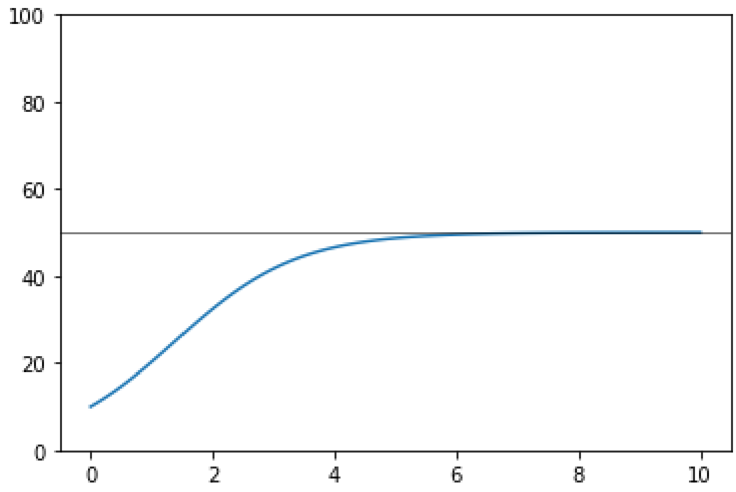
We can look at logistic growth as a mathematical equation. Population growth rate is measured in number of individuals in a population (N) over time (t). The term for population growth rate is written as (dN/dt). The d just means change. K represents the carrying capacity per individual for a population. The logistic growth equation growth equation assumes that K and r do not chnage over time in a population.

In [12]:

```
def f(N,t,r,K):
    return r * N * (1-N/K)

r = 1
K = 50
N0 = 10
t = np.linspace(0,10,100)

N = odeint(f,N0,t,args = (r,K))
plt.plot(t,N)
plt.ylim([0,100])
plt.axhline(y=K,color='k',linewidth=0.5)
plt.show()
```



In [13]:

```
N0 = 80
t = np.linspace(0,10,100)

N = odeint(f, N0, t, args = (r,K))
plt.plot(t,N)
plt.ylim([0,100])
plt.axhline(y=K,color='k',linewidth=0.5)
plt.show()
```