Question 3

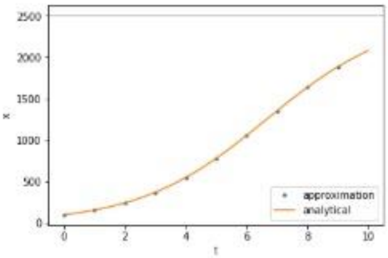
In [1]:
```
import numpy as np
import math
from numpy import *
from sympy import *
from pylab import *
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from sympy.interactive import printing
printing.init_printing(use_latex=True)
```

In [2]:
```
t = arange(0,10,1)
# parameters
r = 0.478
K = 2500
# initial condition
x0 = 100

# Let's define the right - hand side of the differential equation
# It must be a function of the dependendent variable (x) and of the time (t), even if time does not appear explicit;y.
# this is how you define a function:
def f(x,t,r,K):
    # in python, there are no curling braces '{}' to start or end a function, nor any special keyword:
    # the block is defined by leading spaces (usually 4)
    # arithmetic is done the same as in other languages: +, -, *, /
    return r*x*(1-x/K)

# call the function that performs intergration
# the order of the arguments as below: the derivative function, the initial condition, the points where you want the solution
# and the list of the parameters
x = odeint(f,x0,t,(r,K))

#plot the solution
plt.plot(t,x,'.')
plt.xlabel('t')
plt.ylabel('x')
t = arange(0,10,0.01)
# plot analytical solution
# notice that 't' is an array when you do an arithmetic operation
# with an array, it is the same as doing it for each element.
plt.plot(t,K*x0*exp(r*t)/(K+x0*(exp(r*t)-1)))
plt.axhline(y=K,color='k',linewidth=0.5)
plt.legend(['approximation','analytical'],loc = 'best') # draw legend
plt.show()
```



Question 4

In [3]:
```
t,k = Symbol('t'),Symbol('k')
P = Function('P')(t)
diffeq = Eq(P.diff(t)-k*P,0)
sol = dsolve(diffeq)
P0 = Symbol('P0')
constants = solve([sol.subs([(P,P0),(t,t)]),sol.subs([(P,5*P0),(t,t+1)])])
sol = sol.subs(k,constants[1][k])
sol
```

In [22]:
```
c1 = solve([sol.subs([(P,P0),(t,0)]),sol.subs([(P,5*P0),(t,1)])])
c1
```

Out[22]: $\{C_1 : P_0\}$

In [24]:
```
sol = sol.subs(c1)
sol
```

Out[24]: $P(t) = P_0 e^{0.405465108108164t}$

In [25]:
```
solve([sol.subs(P,5*P0)],t)
```

Out[25]: $\{t : 2.70951129135146\}$

Example 02: Plot the function $y = P_0 e^{0.405465108164}, y_0 = 20$

In [27]:
```
t = arange(0,10,0.5)
P0 = 20
y = 20*exp(0.405465108108164*t)
plt.plot(t,y)
plt.xlabel('time')
plt.ylabel('Number of bacteria')
plt.title('Law of Natural Growth')
plt.show()
```