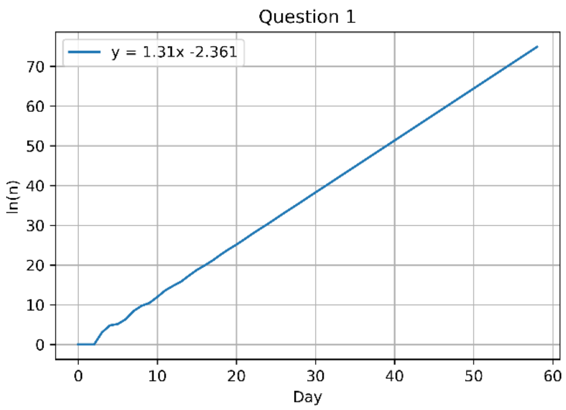
Assignment 6

1) Estimation of population rate of increase

a) The exponential rate of population increase (μ) is $1.310/day$.

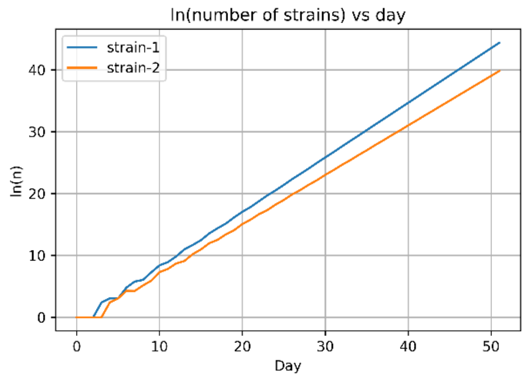
b) Doubling time = $\frac{ln2}{\mu}$ = 0.529 day



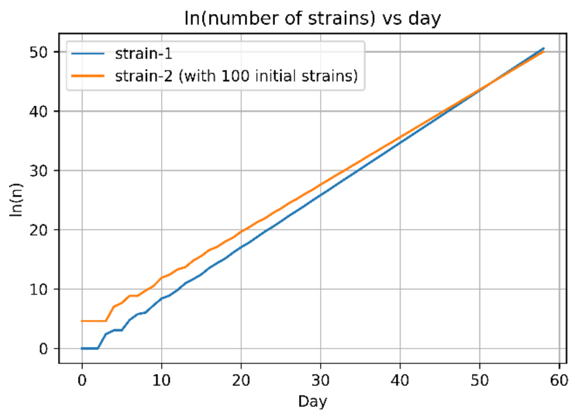
Note: Code for the simulation is attached on the last page.

2) Early vs. late reproduction

a) Growth rate of strain - 1 = $0.884/day$.
Growth rate of strain - 2 = $0.801/day$.
Strain-1 increases faster in the number of individuals than strain-2.



b) It takes around 45 days for the faster strain to reach 50% of the population.



c) The strain with early reproduction grows fasters because early progeny itself also begins to reproduce earlier, etc. For example, each progeny that was produced by strain 1 on day 3 will already have produced 10 progeny on its own by day 6 (when the bulk of reproduction of strain 2 occurs). Thus each additional progeny produced on day 3 contributes ~10 times more to population growth than progeny produced on day 6. This means that mutations that promote early reproduction will generally be selected by evolution, even if they have negative effects on reproduction later in life. Of course, this is a simplified view, and reality is much more complex. For example, we assume that environmental conditions are constant and that individual do not interact with each other. More complex models can discuss many of these details

In [2]:

```
# Importing libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

In [3]:

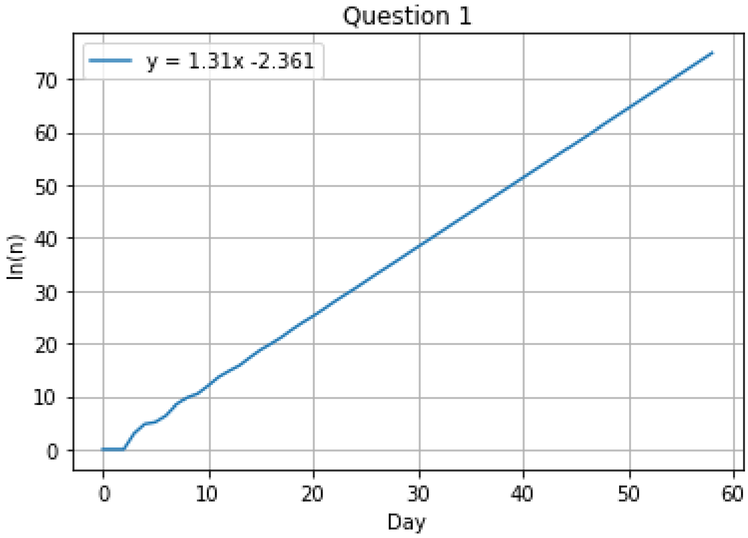
```
# This is for linear regression
def objective(x, a, b):

    return a * x + b
```

In [4]:

```
# This is a simulation function. It takes to input argument d and day_reproducti
on_mapping.
# d is the initial condition, for example if the number of strains at day = 1 is
3 then d = {1 : 3}.
# day_reproduction_mapping is the life history of strain. For example day_reprod
uction_mapping = {3 : 20 , 4 : 100 , 5 : 50} for question 1
def simulation(d , day_reproduction_mapping , print_ = False):

    total_number = [d[1]]
    day = [1]


    for i in range(2 , 60):

        d_new = dict()

        for key in day_reproduction_mapping.keys():

            d_new[1] = d_new.get(1 , 0) + d.get(key , 0) * day_reproduction_mapp
ing[key]

        for key , value in d.items():

            d_new[key+1] = d_new.get(key+1 , 0) + value

        d = d_new

        if print_:
            print("day = " ,i, ",", np.sum(list(d_new.values())) , d)
        total_number.append(np.sum(list(d_new.values())))
        day.append(i)

        if 7 in d_new.keys():

            del d[7]

    return day ,  total_number
```

In [5]:

```
# Question 1 Estimation of population rate of increase
day , total_number = simulation({1:1.0} , {3 : 20 , 4 : 100 , 5 : 50})

# Fitting the straight line between lg n and day
popt, _ = curve_fit(objective, day , np.log(total_number))
a, b = popt
y_new = objective(np.array(day), a, b)

s = "y = " + str(round(a , 3)) + 'x ' + str(round(b , 3))
plt.figure()
plt.plot(list(np.log(total_number)), label = s)
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('Question 1')
plt.grid()
plt.legend()
plt.savefig('fig1.png' , dpi = 600 , facecolor = 'white')
```



In [6]:

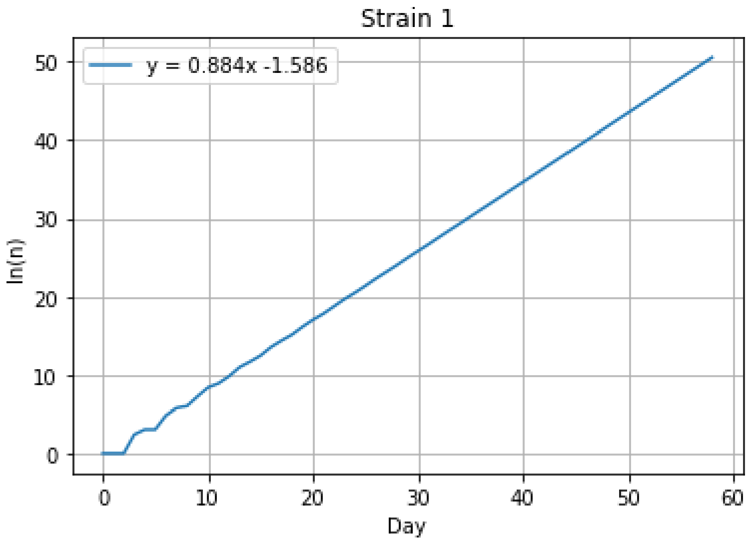
```
# Question 2: Early vs. late reproduction
d = {1:1.0}
day_1 ,  total_number_1 = simulation(d , {3 : 10 , 4 : 10}) # Strain 1
day_2 ,  total_number_2 = simulation(d , {4 : 10 , 5 : 10 , 6 : 50}) # Strain 2
day_3 ,  total_number_3 = simulation({1:100.0} , {4 : 10 , 5 : 10 , 6 : 50}) # S
train2 with different initial condition
```

In [9]:

```
# Plotting for strain 1
popt, _ = curve_fit(objective, day_1 , np.log(total_number_1))
a_1, b_1 = popt
y_new = objective(np.array(day), a, b)
s = "y = " + str(round(a_1 , 3)) + 'x ' + str(round(b_1 , 3))
plt.figure()
plt.plot(list(np.log(total_number_1)), label = s)
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('Strain 1')
plt.grid()
plt.legend()
```

Out[9]:

```
<matplotlib.legend.Legend at 0x12d22523310>
```

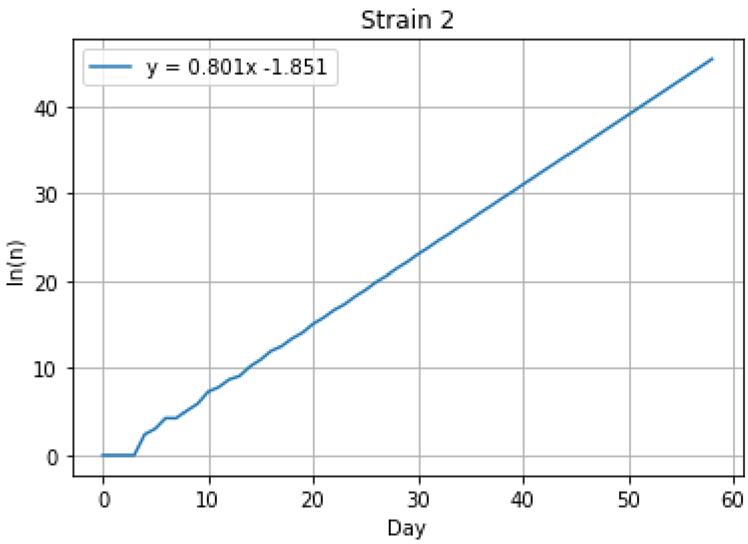


In [10]:

```
popt, _ = curve_fit(objective, day_2 , np.log(total_number_2))
a_2, b_2 = popt
y_new = objective(np.array(day_2), a_2, b_2)
```

In [11]:

```
# Plotting for Strain 2
s = "y = " + str(round(a_2 , 3)) + 'x ' + str(round(b_2 , 3))
plt.figure()
plt.plot(list(np.log(total_number_2)), label = s)
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('Strain 2')
plt.grid()
plt.legend()
plt.savefig('Fig1.png' , dpi = 600 , facecolor = 'white')
```



In [12]:

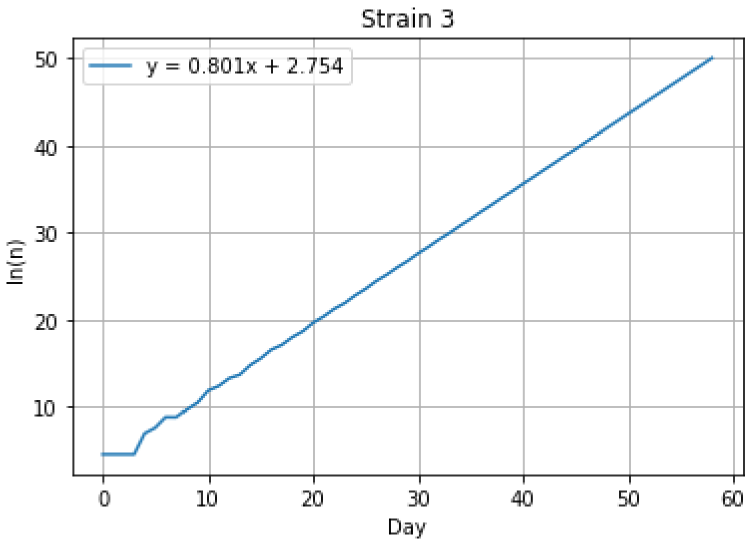
```
popt, _ = curve_fit(objective, day_3 , np.log(total_number_3))
a_3, b_3 = popt
y_new = objective(np.array(day), a_3, b_3)
```

In [13]:

```
# Plotting for strain 3
s = "y = " + str(round(a_3 , 3)) + 'x + ' + str(round(b_3 , 3))
plt.figure()
plt.plot(list(np.log(total_number_3)), label = s)
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('Strain 3')
plt.grid()
plt.legend()
```

Out[13]:

```
<matplotlib.legend.Legend at 0x12d248dc9a0>
```

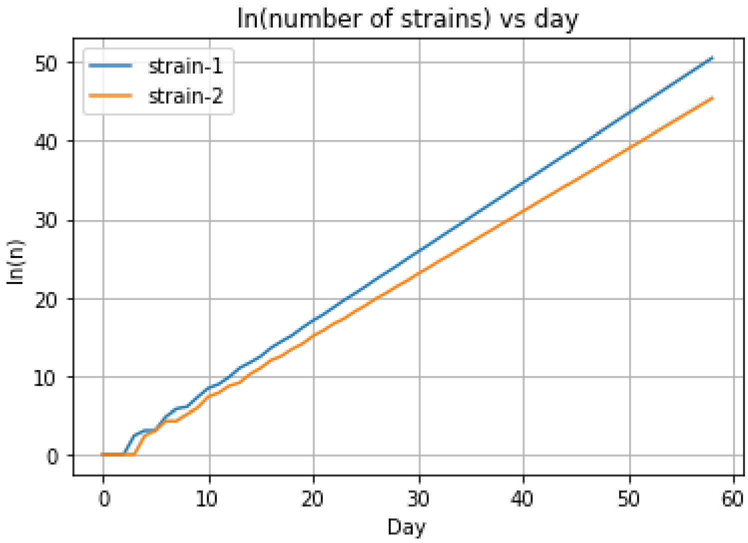


In [14]:

```
plt.figure()
plt.plot(np.log(total_number_1) , label = 'strain-1')
plt.plot(list(np.log(total_number_2)) , label = 'strain-2')
plt.legend()
plt.grid()
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('ln(number of strains) vs day')
```

Out[14]:

```
Text(0.5, 1.0, 'ln(number of strains) vs day')
```

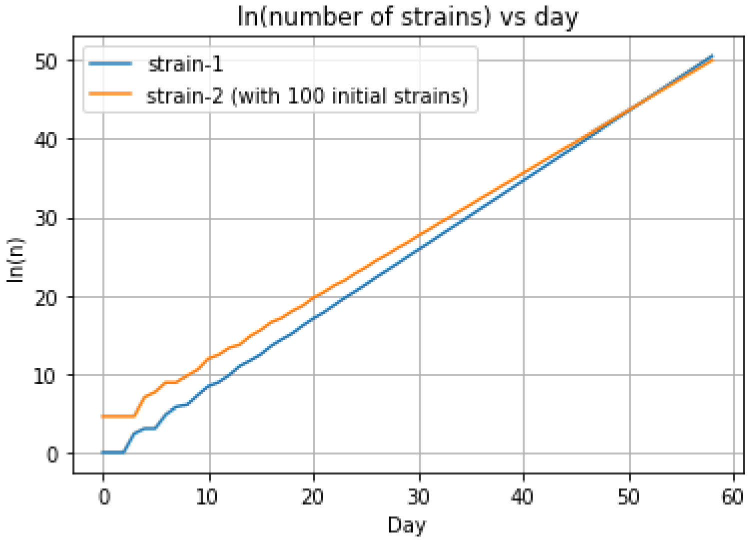


In [15]:

```
plt.figure()
plt.plot(np.log(total_number_1) , label = 'strain-1')
plt.plot(list(np.log(total_number_3)) , label = 'strain-2 (with 100 initial stra
ins)')
plt.legend()
plt.grid()
plt.xlabel('Day')
plt.ylabel('ln(n)')
plt.title('ln(number of strains) vs day')
```

Out[15]:

```
Text(0.5, 1.0, 'ln(number of strains) vs day')
```



In [16]:

```
# Calucating the day when faster growing strain is over 50% of the total populat
ion

for i in range(len(total_number_1)):

    if (total_number_1[i] / total_number_3[i]) > 0.50:

        print("day = " , i + 1)
        break
```

```
day =  45
```