Problem 1(a)

In [ ]:

```python
# Import necessary libraries
from scipy.integrate import odeint
import numpy as np
import matplotlib.pyplot as plt

# Define the model function
def model(
    Rp,
    t,
    S,
    k1=1,
    k2=1,
    Rt=1,
    km1=0.05,
    km2=0.05,
    ):
    """
    Model function to describe the rate of change of Rp over time.

    Parameters:
    Rp : float
        Current value of Rp.
    t : float
        Current time point.
    S : float
        Parameter S affecting the rate.
    k1 : float, optional
        Rate constant for the first term (default is 1).
    k2 : float, optional
        Rate constant for the second term (default is 1).
    Rt : float, optional
        Total concentration (default is 1).
    km1 : float, optional
        Michaelis constant for the first term (default is 0.05).
    km2 : float, optional
        Michaelis constant for the second term (default is 0.05).

    Returns:
    dRpdt : float
        Derivative of Rp with respect to time.
    """
    # Calculate the rate of change of Rp
    dRpdt = (k1*S*(Rt-Rp)/(km1+Rt-Rp)) - k2*Rp/(km2+Rp)

    # Return the derivative
    return dRpdt
```

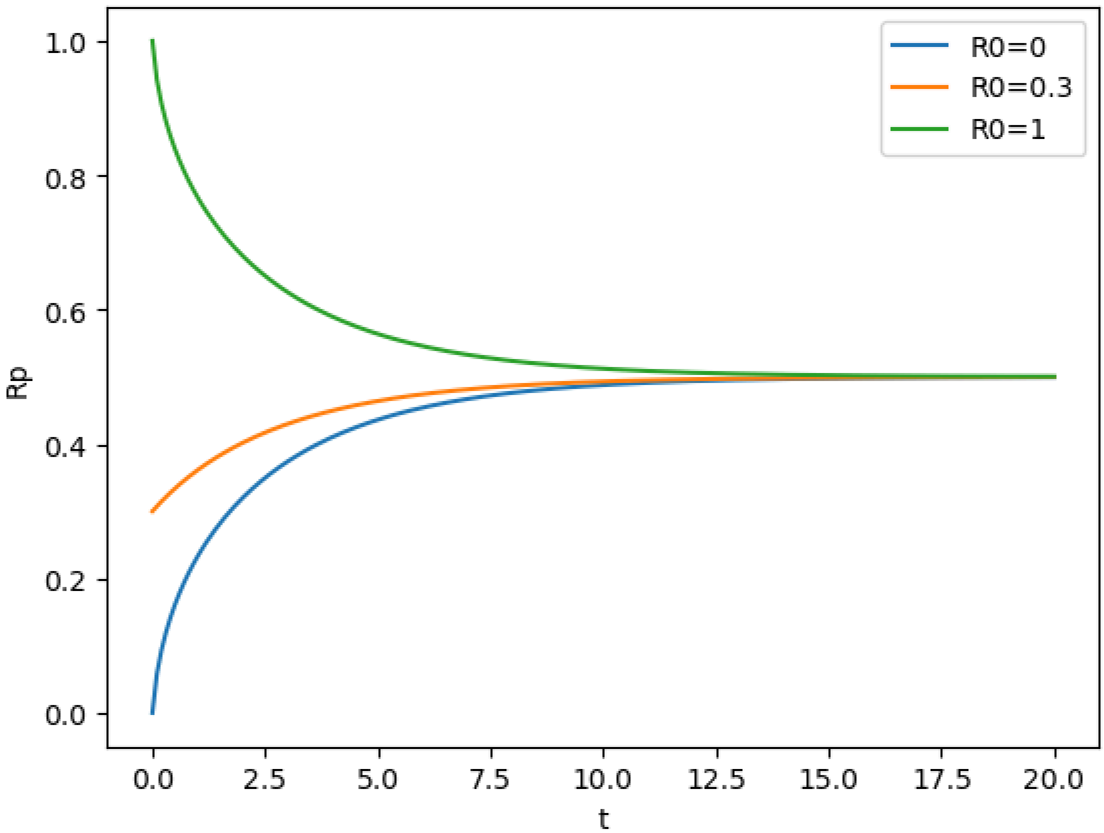
In [3]:

```python
S = 1                                       # Set parameter S
Rp0 = [0,0.3,1]                             # Initial conditions for Rp
t = np.linspace(0,20,200)                   # Time points for simulation
```

```
result = odeint(model,Rp0,t,args=(S,)) # Integrate the ODEs

# Plot the results
fig,ax = plt.subplots()
ax.plot(t,result[:,0],label='R0=0')
ax.plot(t,result[:,1],label='R0=0.3')
ax.plot(t,result[:,2],label='R0=1')
ax.legend()
ax.set_xlabel('t')
ax.set_ylabel('Rp')
```

Out[3]: `Text(0, 0.5, 'Rp')`



Problem 1(b)

$$\frac{dR_p}{dt} = \frac{k_1 S(R_t - R_p)}{k_{m1} + R_t - R_p} - \frac{k_2 R_p}{k_{m2} + R_p}$$

Steady state when $\frac{dR_p}{dt} = 0$

$\implies 0 = \frac{k_1 S(R_t - R_p)}{k_{m1} + R_t - R_p} - \frac{k_2 R_p}{k_{m2} + R_p}$

$\implies \frac{k_1 S(R_t - R_p)}{k_{m1} + R_t - R_p} = \frac{k_2 R_p}{k_{m2} + R_p}$

Plugging in constants: $k_1 = 1$ $k_2 = 1$ $S = 1$ $R_t = 1$ $k_{m1} = 0.05$ $k_{m2} = 0.05$

$$\frac{1 \times 1 \times (1 - R_p)}{0.05 + 1 - R_p} = \frac{1 \times R_p}{0.05 + R_p}$$

$$\implies \frac{1 - R_p}{1.05 - R_p} = \frac{R_p}{0.05 + R_p}$$

$$\implies (1 - R_p)(0.05 + R_p) = R_p(1.05 - R_p)$$

$$\implies 0.05 + .95R_p - R_p^2 = 1.05R_p - R_p^2$$

$$\implies 0.05 = 0.1R_p$$

$$\implies R_p^s s = 0.5$$

Problem 2

In [4]:
```
# Problem 2i.
import numpy as np

# Define matrix A
A = np.array([[1, 2],
              [3, 4],
              [5, 6]])

# Define matrix B
B = np.array([[1, 0],
              [0, 1]])

# Get the dimensions of A and B
A_shape = np.shape(A)
B_shape = np.shape(B)

print("Shape of A:", A_shape)
print("Shape of B:", B_shape)
```

```
Shape of A: (3, 2)
Shape of B: (2, 2)
```

In [5]:
```
# Problem 2ii.

C = A + 1

print("Matrix C (A + 1):\n", C)
```

```
Matrix C (A + 1):
 [[2 3]
 [4 5]
 [6 7]]
```

In [6]:
```
A + B  # This will raise an error due to incompatible shapes
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[6], line 1
----> 1 A + B  # This will raise an error due to incompatible shapes

ValueError: operands could not be broadcast together with shapes (3,2) (2,2)
```

In [7]:
```
# Problem 2iii.
D = C @ B

print("Matrix D (C @ B):\n", D)
```

```
Matrix D (C @ B):
 [[2 3]
 [4 5]
 [6 7]]
```

In [8]:
```
G = C.T

G * B # This will raise an error due to incompatible shapes
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[8], line 3
      1 G = C.T
----> 3 G * B # This will raise an error due to incompatible shapes

ValueError: operands could not be broadcast together with shapes (2,3) (2,2)
```

In [11]:
```
# Problem 2iv.

print("This is the result of matrix multiplication:\n")
print(G @ C @ B)   # Matrix multiplication

print("\nThis is the result of element-wise multiplication:\n")
print(G @ C * B)   # Element-wise multiplication
```

```
This is the result of matrix multiplication:

[[56 68]
 [68 83]]

This is the result of element-wise multiplication:

[[56  0]
 [ 0 83]]
```

Problem 3

Variable	Meaning	Dimension	Unit
V	Reactor volume	Volume	L

Variable	Meaning	Dimension	Unit
$[S]$	Substrate concentration in reactor	Amount / Volume	mol / L
$[S]_o$	Substrate concentration in feed	Amount / Volume	mol / L
t	Time	Time	hr
K	Volumetric flow rate into reactor	Volume / Time	L / hr
$V_{\max}$	Maximum reaction rate	Amount / Time	mol / hr
K_m	Michaelis constant	Amount / Volume	mol / L