1. Import the required Python modules: NumPy, Matplotlib, and SciPy's optimization library.
2. Define input data as arrays t (time) and u (voltage across a capacitor)
3. Define the model function that will fit the data. The expon_func function has an exponential trend, and takes as input the time variable t and parameters A and B. The function returns the output variable UC that represents the voltage across a capacitor in an RLC circuit.

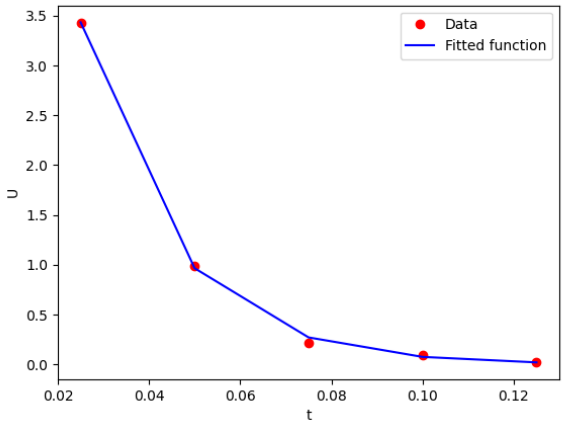
a.
```
def expon_func(t, A, B):
    return A * np.exp(B * t)
```

4. Fit the model function to the data using curve_fit from SciPy's optimization library. The curve_fit function optimizes the values of the parameters to minimize the difference between the model function and the data points. The initial guess for the parameters p0 is set to (12, 0.05).

a. The p0 parameter is set to (12, 0.05), which specifies an initial guess of A = 12, B = 0.05. These values are chosen based on a visual inspection of the data.

b.
```
x, y = opt.curve_fit(expon_func, t, u, p0=(12, 0.05))
```

5. Send the acquired data back to the expon_func to calculate the best fit for Uc(t)
6. Plots the given points and the best fit.
7. Calculate RC
8. Prints RC value.



After comparing the data points and the fitted function it is accurate.

```
RC = 0.0197186201386328
```

RC value