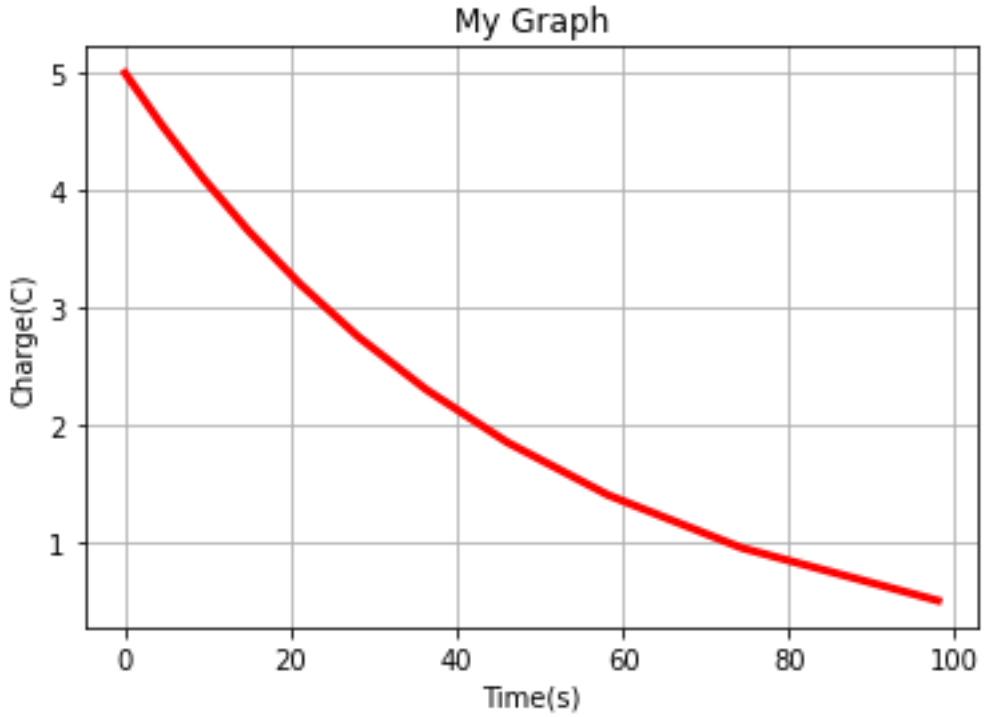
EULER METHOD

```
import matplotlib.pyplot as plt
import numpy as np

def f(q):
 R = 10
 C = 5

 y = (-R * C/ q)

 return y

def my_plot(x, y):
 plt.plot(x, y, linewidth = 3, color = 'r')
 plt.grid(True)
 plt.xlabel('Time(s)')
 plt.ylabel('Charge(C)')
 plt.title('My Graph')
 plt.show()
 plt.clf()

q0 = 5
t0 = 0
h = -0.45
'''
for i in range(10):
 t1 = t0 + (h * f(q0))
 q1 = q0 + h

 print(q1, t1)

 q0 = q1
 t0 = t1
'''
n = 11
t = [0]*n
q = [0]*n

t[0] = t0
q[0] = q0

for i in range (n-1):
 t[i+1] = round(t[i] + h * f(q[i]), 2)
 q[i+1] = round(q[i] + h, 2)

 print(q, t)

my_plot(t, q)
print("\nThe time required is", t[n-1], "sec")
```

OUTPUT –

```
[5, 4.55, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0, 4.5, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[5, 4.55, 4.1, 0, 0, 0, 0, 0, 0, 0, 0] [0, 4.5, 9.45, 0, 0, 0, 0, 0, 0, 0, 0]
[5, 4.55, 4.1, 3.65, 0, 0, 0, 0, 0, 0, 0] [0, 4.5, 9.45, 14.94, 0, 0, 0, 0, 0, 0, 0]
[5, 4.55, 4.1, 3.65, 3.2, 0, 0, 0, 0, 0, 0] [0, 4.5, 9.45, 14.94, 21.1, 0, 0, 0, 0, 0,
0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 0, 0, 0, 0, 0] [0, 4.5, 9.45, 14.94, 21.1, 28.13, 0, 0,
0, 0, 0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 2.3, 0, 0, 0, 0] [0, 4.5, 9.45, 14.94, 21.1, 28.13,
36.31, 0, 0, 0, 0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 2.3, 1.85, 0, 0, 0] [0, 4.5, 9.45, 14.94, 21.1, 28.13,
36.31, 46.09, 0, 0, 0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 2.3, 1.85, 1.4, 0, 0] [0, 4.5, 9.45, 14.94, 21.1,
28.13, 36.31, 46.09, 58.25, 0, 0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 2.3, 1.85, 1.4, 0.95, 0] [0, 4.5, 9.45, 14.94, 21.1,
28.13, 36.31, 46.09, 58.25, 74.32, 0]
[5, 4.55, 4.1, 3.65, 3.2, 2.75, 2.3, 1.85, 1.4, 0.95, 0.5] [0, 4.5, 9.45, 14.94, 21.1,
28.13, 36.31, 46.09, 58.25, 74.32, 98.0]
```




```
The time required is 98.0 sec
```