GNG1106 – Fundamentals of Engineering Computation
Course Project

RCL Electric Circuit Calculator

Viktor Stojanovic (300137255)
Muhammad Owais Tabassum (0300137880)

Date: December 8, 2019

1 Problem Identification and Statement

Problem: Determine the resistor value from the following user input: component values L and C, the battery voltage, the dissipation time t_d, and the percentage of original charge P_c to reach within the dissipation time. Plot a charge over time functions with the determined variables. Store variables in text file.

Solution: The software will use the following equations to eventually find the value for resistance and plot a charge over time graph:

Determining the root of function g(R) will find the R value required in the function q(t).

$$g(R) = e^{-Rt_d/(2L)} - p_c$$

$$\left(R = \frac{-2L\ln(p_c)}{t_d} \right)$$

In order to plot function q(t) R is required.

$$q(t) = q_0 e^{-Rt/(2L)} \cos\left[\left(\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} \right) t \right]$$

The method to find R will be the Bisection method, a recursive method that approximates the root's value. This method involves a loop where an interval is getting smaller by halves, until the value of the root, within error, is determined. An interval is determined to have a root within it when the factor of its upper bound and lower bound is less than zero. The interval will keep getting smaller until the factor of the bounds is less than a number previously determined to be small enough to be considered zero, and the midpoint of that interval will be the root. The initial bounds of the method will be 0 to R_{Max}.

$$R_{\max} = \sqrt{\frac{4L}{C}}$$

The method to plot the will be the PLplot library, the library used in lab 6. The plotting functions from lab 6 will be repurposed to plot function q(t).

The method to store user data will be a text file. There will be sets of data (Resistance, L, C...) stored in the text file that will correlate to the sets of data the user will store. The file will be opened near the beginning of the program to obtain previous user data. When the user would like to store information, the text file will open in writing mode and add the information to the end of the file. The program will only read the first 5 sets of data as to limit the storage of data to 5. An added function could be to delete a data set. This would involve the user inputting which data set to delete (1-5), and having the program rewrite the text file skipping over that specific data set. This would shift data sets that come after it by one to the left.

2 Gathering of Information and Input/Output Description

2.1 SubSection 1

List of variables

V: Voltage drop (Volt)
R: Resistance in circuit (Ohms)
L: Inductance in circuit (Henrys)
i: Current in circuit (Amperes)
q: charge in circuit (Coulombs)
C: Capacitance in circuit (Farads)

The flow of current, *i*, through the resistor causes a voltage drop (V_R). The V determined will generally be in units of volts. The value for this voltage drop can be derived using this equation, given by

$$V\,r = iR$$

The equation below will determine the voltage drop with respect to inductance. Inductance is the resistance of the change in current. VL will be the voltage drop due to this effect, given by

$$V\,L = L\frac{di}{dt}$$

This equation will determine the voltage drop across the capacitor. This voltage drop is highly dependant on the charge (q) on the capacitor, C is the capacitance and has the units Farad (F). the units for VC will be in volts.

$$V\,C = \frac{q}{C}$$

According to Kirchhoff's second law, the sum of all the voltage drops is equivalent to zero. Therefore, our new equation after the switch is closed is below.

$$V\,L + V\,r + V\,C = 0$$
$$L\frac{di}{dt} + Ri + \frac{q}{C} = 0$$

According to the equation below, the current is related to the charge. Therefore, the equation to calculate current can be incorporated into the previous equation.

$$i = \frac{dq}{dt}$$

The new equation after the substitution is the following

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = 0$$

Using calculus, it is possible to solve the equation above like this.

$$q(t) = q_0 e^{-Rt/(2L)} \cos\left[\left(\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}\right) t\right]$$

2.2 SubSection 2

This equation represents the amplitude of the oscillations within the graph of the cosine function.

$$q(t) = q_0 e^{-Rt/(2L)}$$

This equation is used to calculate the amplitude at a dissipation time.

$$q(t_d) = p_c q_0 = q_0 e^{-Rt_d/(2L)}$$
$$p_c = e^{-Rt_d/(2L)}$$

The equation below will derive the value for resistance (R) at the root.

$$g(R) = e^{-Rt_d/(2L)} - p_c$$

To determine the value for the resistance, there are many steps and equations involved that must be satisfied. Using the listed equations and the root it is possible to find the value of the resistance (R) with the equation of g(R).

2.3 SubSection 3

There is a method for finding roots called the bisection method, it is also known as binary chopping, interval halving or Balzano's Method. essentially this method is used to determine whether or not the function will have a root within a specific interval. This method is done in a loop where the interval is getting smaller by halves, until the value of the root is determined. If the function is passing through the x-axis, the factor of the upper bound and lower bound of the interval is smaller than zero. When the value is greater than zero this will mean that the function is not passing through or on the x-axis. The new interval is determined by chopping the current interval in half using using [f(final bound)+f(initial bound)]/2. This new interval is at the midpoint of the previous one. This loop will continue until it is determined when the function has roots and where it does not. This method is appropriate as it will give a more accurate value for the roots.

2.4 Input and Output

The user inputs (L, C, the battery voltage, the dissipation time t_d, and the percent of original charge p_c to reach within dissipation time) are required by the software to determine the value of R. The file stores values for the user to use, or the user may enter new values. When the value of t_d is too small, the R value cannot be found because it will be too large; In these cases, the software will print a message. Also, during the execution of this program, if the frequency of the oscillation is too large, for example, 50 cycles within time t_d, so that $50/f < t_d$, the software

will not have an accurate plot graph and will print the message, “Frequency too high; plot may be distorted”.

- 3 Test Cases and Design

- 3.1 Test Cases

This table will outline extreme cases that might cause our program to react differently.

Case #	L(Henrys)	C (Farads)	Voltage (volts)	t_d (s)	p_c	R (Ohms)	Functionality
1	91	12	6	1	0.9	19.18	X
2	91	12	6	30	0.9	0.64	✓
3	20	12	6	30	0.9	0.14	✓
4	91	4	6	30	0.9	0.64	✓
5	91	12	2	30	0.83	1.13	✓
6	33	2	6	30	0.5	1.52	✓
7	54	7	2	15	0.2	11.59	X
8	2	8	5	25	0.25	0.22	✓

$$\left(R = \frac{-2L\ln(p_c)}{t_d} \right)$$

$$q(t) = q_0 e^{-Rt/(2L)} \cos\left[\left(\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} \right) t \right]$$

In theory, when the resistance (R) value is too large the square root inside the cosine function will be negative and it will not work. Thus, test case 1 does not work because the t_d value is too small causing for the resistance value to be unreasonably large to the point where the value inside the root function will be negative causing for it to be invalid. This is similar for test

case 7 as the L and C values are both very small and the t_d is also quite small. This means that the resistance calculated will still be too large. The same thing happens here as it did for test case 1. The value inside the root function will be negative causing for it to be invalid. The test cases other than 1 and 7 are all valid as the values do not calculate to give a negative number within the root function.

There will be a test case in which the user is asked to store their data in a file. It will test if the data is stored in the file successfully.

3.2 Design

Name of program: *RCL Electric Circuit Calculator*

Main.c

- Contains all functions

EE_Data.txt

- Holds user data obtained from RCL_Electric_Circuit_Calculator.c

Description:

The program reads EE_Data.txt to receive any past data.The user is prompted to use previous data or input their own.

If user uses previous data, selected data with be used to plot function q(t) (Charge over Time).

If user chooses to input their own data, they input: L, C, the battery voltage, the dissipation time t_d, and the percent of original charge p_c to reach within dissipation time into the program when prompted.

The program, using the bisection method, calculates the root R and displays it to the user. The program will plot q(t) (Charge over Time) with the newly acquired data.

The user will be prompted to store data into the file EE_Data.txt. If a data slot is available, they may save it there. Otherwise, they will be prompted to replace an existing data slot.

- ### 3.2.1 Introduction

void main()

void main()

Parameters:

- None

Return value

- None

Logic/Algorithm

This function starts off the program. It will call on other functions and will dictate the order of when a function is used. When this function ends, the program will end.

-

- ### 3.2.2 Functions for Interacting with the User

void getUserData(DATA [], DATA *)

void getUserData(DATA file[], DATA *selected)

Parameters:

- DATA file: Keeps values obtained through the file EE_Data.txt. It is an array that carries up to 5 data sets.
- DATA selected: Contains values that the user chose.

Return value

- None

Logic/Algorithm

The user is presented with all previous user data contained in DATA file. The user will make a choice to select data from the presented data or input it in themselves. Selected data will be put into DATA selected for future use in the program.

void SaveData(DATA [], DATA *)

void SaveData(DATA file[], DATA *selected)

Parameters:

- DATA file: Keeps values obtained through the file EE_Data.txt. It is an array that carries up to 5 data sets.
- DATA selected: Contains values that the user chose.

Return value

- None

Logic/Algorithm

The user is presented with all previous user data contained in DATA file. The user will make a choice to save data they have selected into the DATA file. If DATA file has less than 5 data sets, the user can save the data into a free slot. If the data has 5 data sets, the user will be prompted to replace a data set of their choosing.

double overZero(char [])

double overZero(char var[2])

Parameters:

- char var[2]: This string correlates to a specific data point from the structure.

Return value:

- The specific data point's value given by the user

Logic/Algorithm

This function will be used to get user inputs that are not negative or equal to zero.

- 3.2.3 Functions for Calculations

double qfunc(DATA, double)

double qfunc(DATA parameters, double t)

- **Parameters:**

  - t: specific time the of the charge over time function
  - parameters: specific values inputted by user

Return value:

- Specific charge at a given time

Logic/Algorithm:

Returns the specific charge at a given time using data the user inputted previously and the time. It will use the q(t) equation.

double Rfunc(DATA, double)

double Rfunc(DATA parameters, double R)

Parameters:

- t: specific time the of the charge over time function
- parameters: specific values inputted by user

Return value:

- Value of g(R) at a given R

Logic/Algorithm:

Returns value of g(R) at a given resistance using data the user inputted previously and the time. It will use the g(R) equation.

double findRoot(double , double, DATA)

double findRoot(double lower, double upper, DATA data)

Parameters:

- lower: Lowest x value of interval (0)

- upper: Lowest x value of interval ($R_{max} = \sqrt{\frac{4L}{C}}$)
- data: The values the user chose

Return value

- Root of a function at a given interval

Logic/Algorithm:

Returns root at a given interval of a function. The method to find the root is the bisection method using the lower and upper bound given. It will calculate the midpoint of the interval. This function will use qfunc() or Rfunc() to a negative factor between the upper and lower bounds. Once this is found it will determine on which side of the midpoint the root lies on and adjust the interval accordingly. Once the factor of the function of the interval's bonds is less than a certain chosen value, the program will detect a root in the midpoint of this interval.

double findUpper(DATA)

double findUppert(DATA parameter)

Parameters:

- parameter: Data the user has chosen.

Return value

- Upper value of the interval used to find a root in function g(R)

Logic/Algorithm: This function calculates the upper bound using the L and C values determined from DATA parameter.

int validate(DATA)

int validate(DATA data)

Parameters:

- data: Data the user has selected

Return Value:

- Returns 0 if user data will not yield an R value. Returns 1 if user data will yield an R value.

Logic/Algorithm:

This function is used to validate if the values that the user inputted meet the criteria. The equation used in this function is as follows...

$$\frac{1}{LC}-\left(\frac{R}{2L}\right)^2$$

If the value calculated from this equation is positive, the program will continue to work as it is possible to take the square root of a positive number. If the value is negative the program will stop working because it is not possible to take the square root of a negative number.

Function is derived from $q(t)=q_0e^{-Rt/(2L)}\cos\left[\left(\sqrt{\frac{1}{LC}-\left(\frac{R}{2L}\right)^2}\right)t\right]$.

- 3.2.4 Functions for Plotting

void plotFunc(double, double, DATA)

void plotFunc(double begin, double end, DATA data)

Parameters:

- begin: First x value in the function to be plotted
- end: Last x value in the function to be plotted
- data: Contains all user data

Return Value:

- none

Logic/Algorithm:

Calculates the points (x and y values) on a function within the interval given in the parameter and puts them in 2 separate arrays respectively. It sends these arrays to a plotting function to plot the points calculated.

void plot(int, double*, double*)

- void plot(int n, double *xPtr, double *yPtr)

Parameters:
- n: The number of points in the arrays, which is 100.
- xPtr: This is a pointer to the x values that signify the t (horizontal axis).
- yPtr: This is a pointer to the y values that signify the q(t) (vertical axis).

Return Value:
- none

Logic/Algorithm:

Initialises the plot. The following values in the referenced structure are used to set up the plot x[0], x[n-1] - assume that x values are sequential miny, maxy - vertical axis range (add 10% to min/max value) Sets up white background and black for ground colors.

Then plots the curve accessed using xPtr and yPtr.

double getMin(double *array, int n)

double getMin(double *array, int n)

Parameters:
- double *array: Contains all the points for function q(t).
- int n: Amount of points through the interval of q(t).

Return Value:
- Returns the smallest point on the function q(t).

Logic/Algorithm:

This function is used to find the minimum value for q(t) function.

double getMax(double *array, int n)

double getMax(double *array, int n)

Parameters:
- double *array: Contains all the points for function q(t).
- int n: Amount of points through the interval of q(t).

Return Value:

- Returns the largest point on the function q(t).

Logic/Algorithm:

This function is used to find the maximum value for q(t) function.

-

- 3.2.5 Functions for Files

void getFileData(DATA[])

void getFileData(DATA data[])

Parameters:

- data[]: Structure DATA keeps values the user has inputted. It is an array that carries up to 5 data sets.

Return value

- None

Logic/Algorithm

This function opens file EE_Data.txt and extracts any past data from it. It will use fread to transfer all data into the array arr. Each index of data is representative of a data set.

void writeFileData(DATA[])

void writeFileData(DATA data[])

Parameters:

- data[]: Structure DATA keeps values the user has inputted. It is an array that carries up to 5 data sets.

Return value

- None

Logic/Algorithm

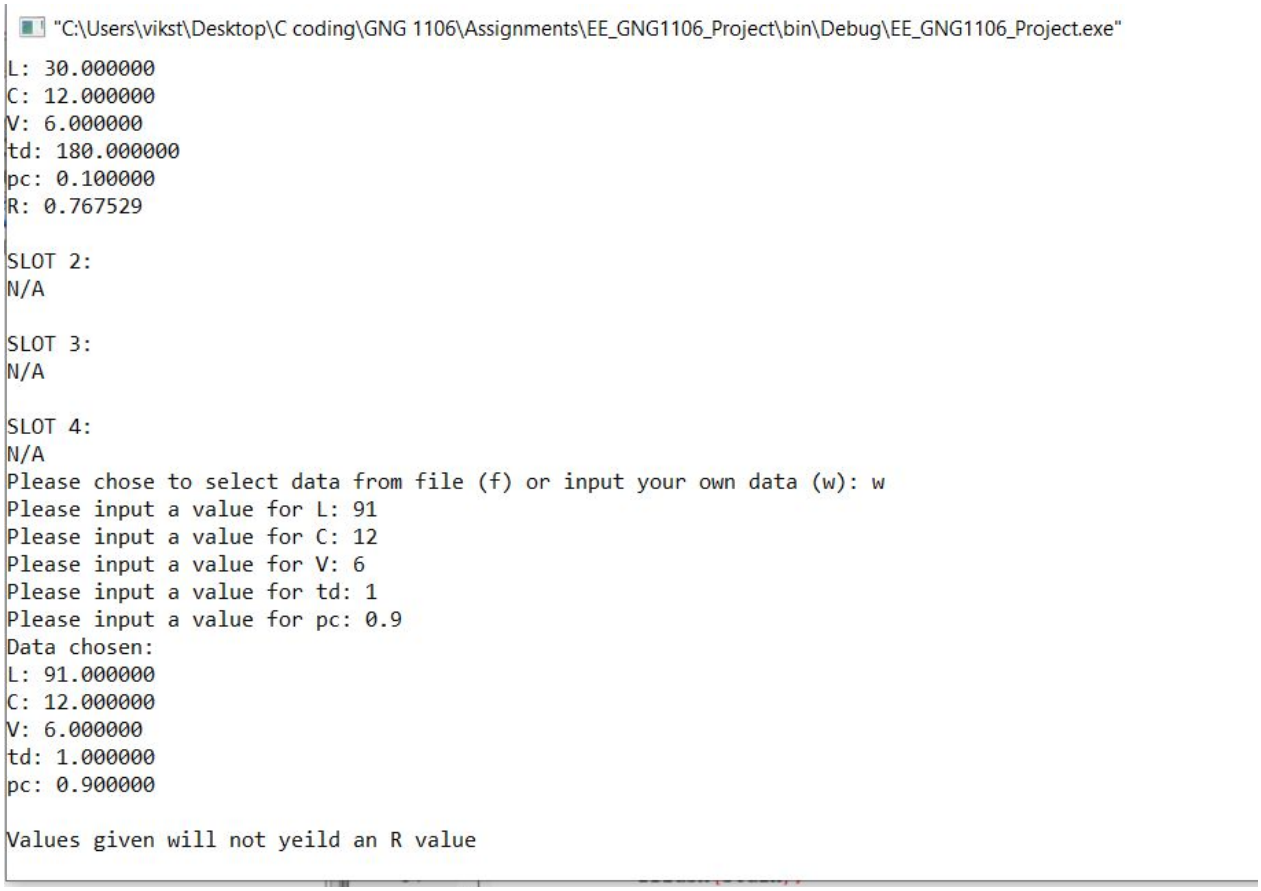
This function opens file EE_Data.txt and rewrite the file with the values in arr[]. It will use fwrite to transfer all data into the file. Each index of data is representative of a data set.

- 4 Implementation

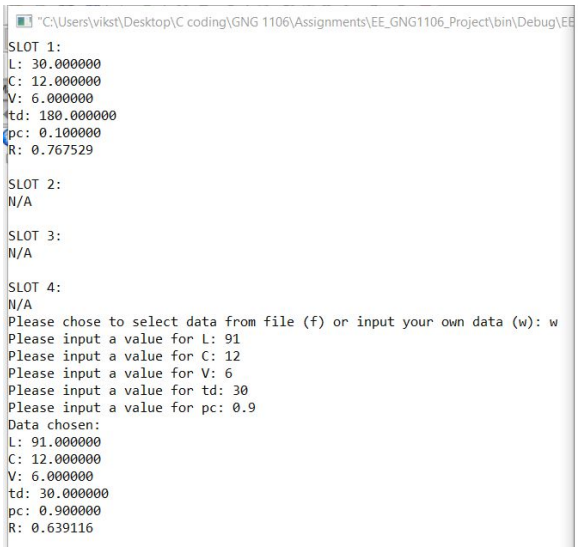
See the Main.c in project EE_GNG1106_Project.cbp

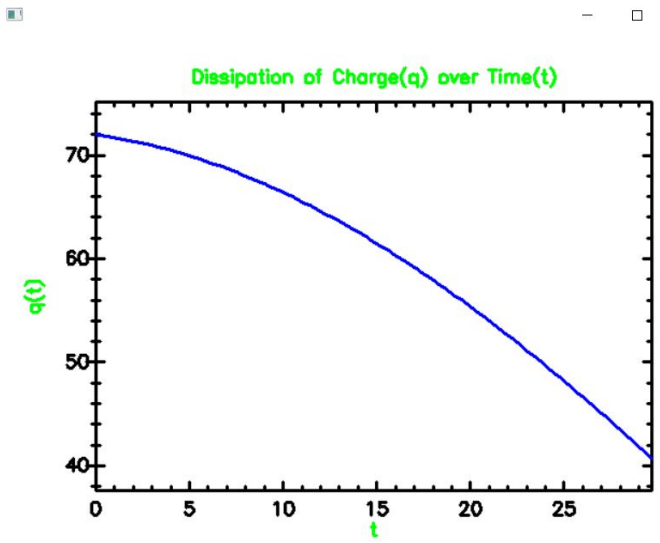
- 5 Software Testing and Verification

Case 1:

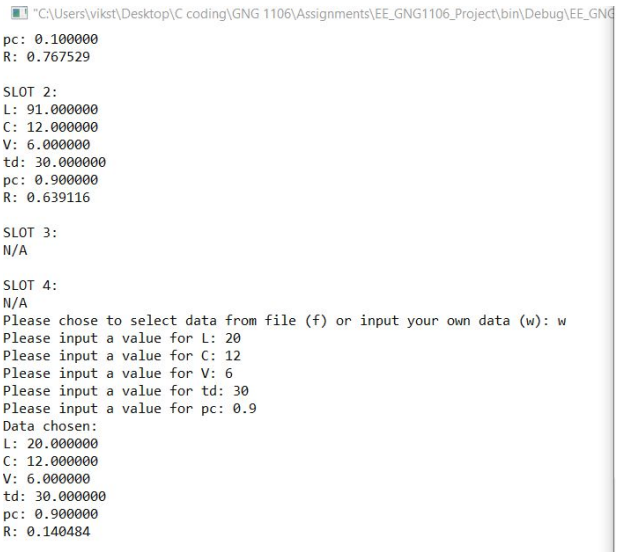


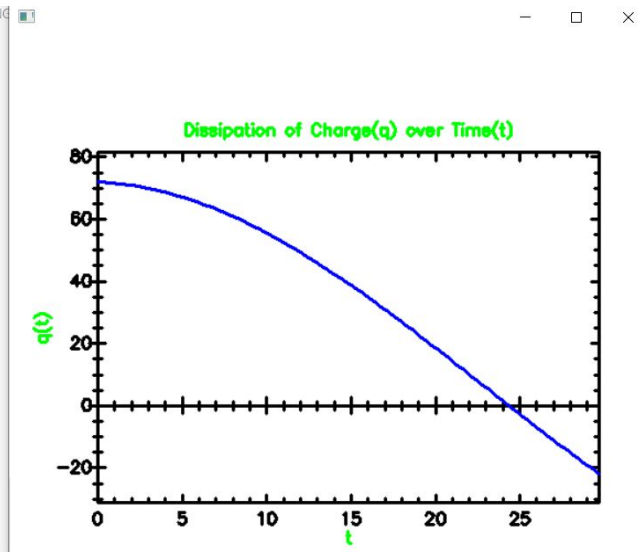
Case 2:


"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Project\bin\Debug\EE
SLOT 1:
L: 30.000000
C: 12.000000
V: 6.000000
td: 180.000000
pc: 0.100000
R: 0.767529
SLOT 2:
N/A
SLOT 3:
N/A
SLOT 4:
N/A
Please chose to select data from file (f) or input your own data (w): w
Please input a value for L: 91
Please input a value for C: 12
Please input a value for V: 6
Please input a value for td: 30
Please input a value for pc: 0.9
Data chosen:
L: 91.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639116

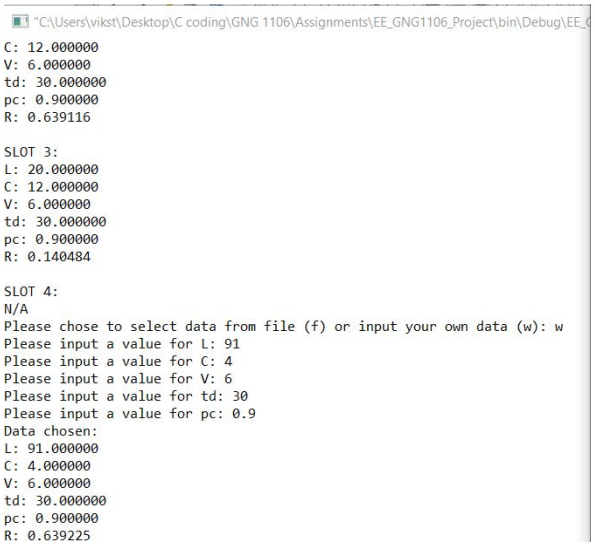

Dissipation of Charge(q) over Time(t)
q(t)
t
70
60
50
40
0
5
10
15
20
25

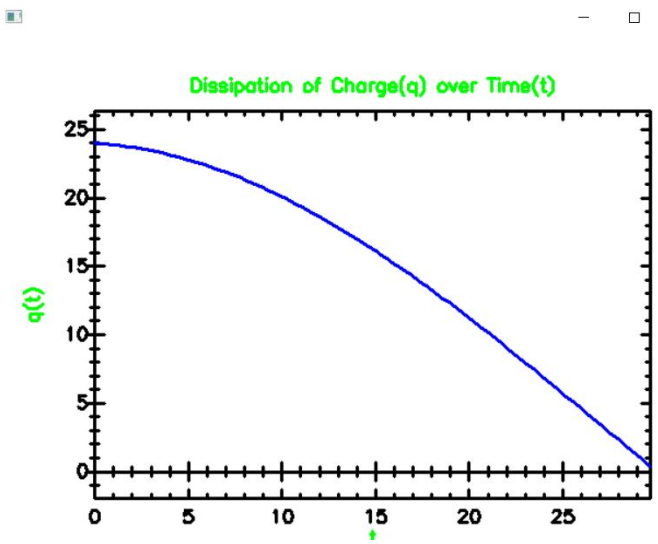
Case 3:


"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Project\bin\Debug\EE_GNG
pc: 0.100000
R: 0.767529
SLOT 2:
L: 91.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639116
SLOT 3:
N/A
SLOT 4:
N/A
Please chose to select data from file (f) or input your own data (w): w
Please input a value for L: 20
Please input a value for C: 12
Please input a value for V: 6
Please input a value for td: 30
Please input a value for pc: 0.9
Data chosen:
L: 20.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.140484

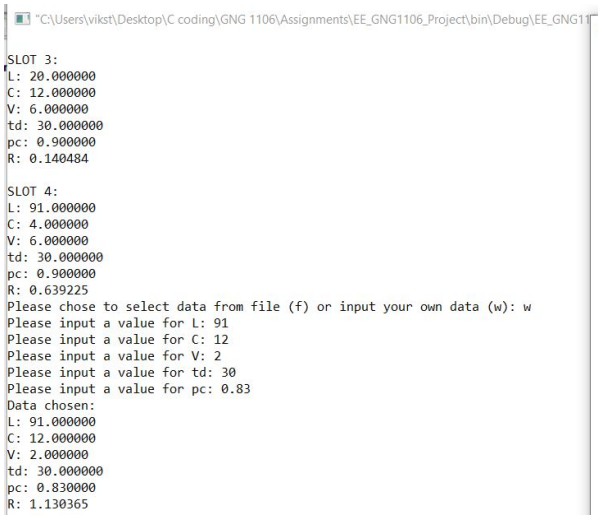

Dissipation of Charge(q) over Time(t)
q(t)
t
80
60
40
20
0
-20
0
5
10
15
20
25

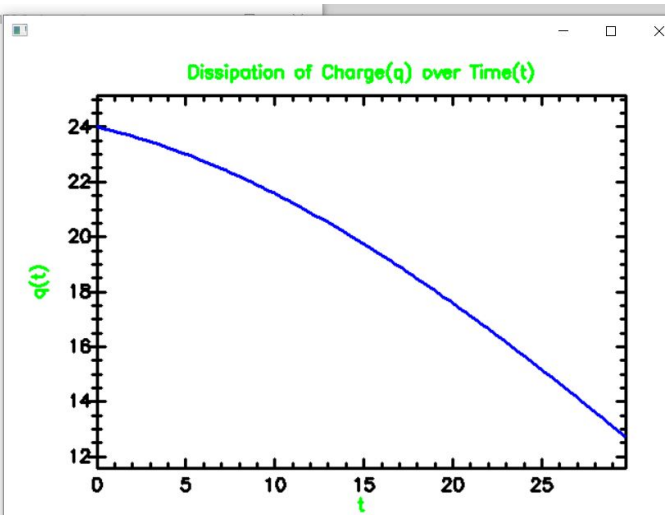
Case 4:


"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Project\bin\Debug\EE_C
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639116
SLOT 3:
L: 20.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.140484
SLOT 4:
N/A
Please chose to select data from file (f) or input your own data (w): w
Please input a value for L: 91
Please input a value for C: 4
Please input a value for V: 6
Please input a value for td: 30
Please input a value for pc: 0.9
Data chosen:
L: 91.000000
C: 4.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639225

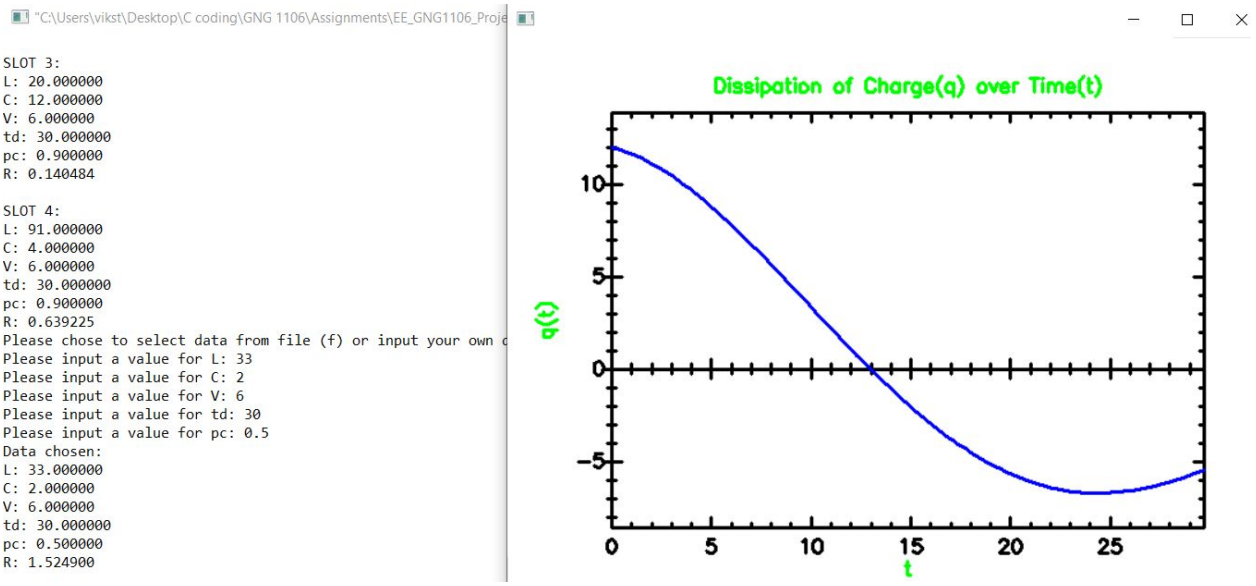

Dissipation of Charge(q) over Time(t)
q(t)
t
0
5
10
15
20
25

Case 5:

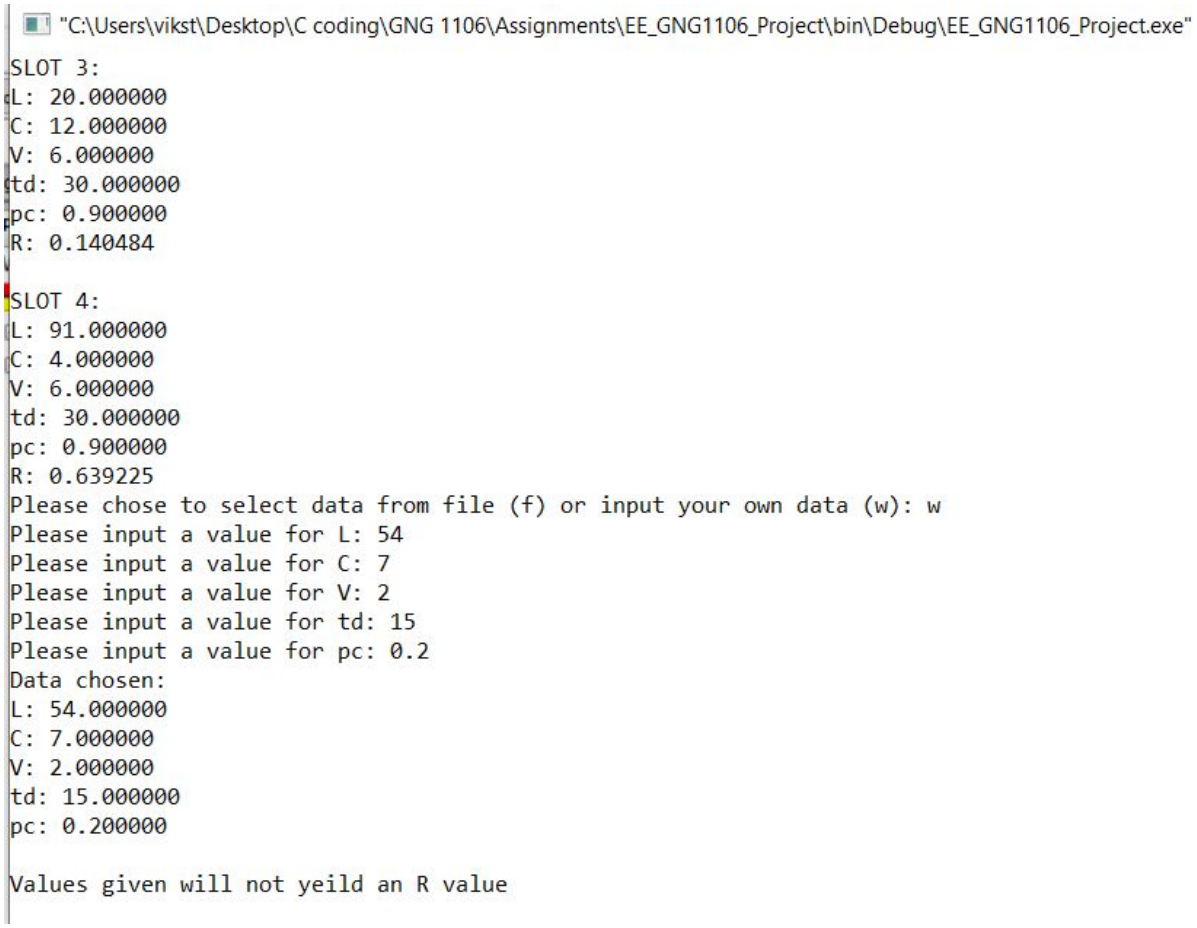

"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Project\bin\Debug\EE_GNG11
SLOT 3:
L: 20.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.140484
SLOT 4:
L: 91.000000
C: 4.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639225
Please chose to select data from file (f) or input your own data (w): w
Please input a value for L: 91
Please input a value for C: 12
Please input a value for V: 2
Please input a value for td: 30
Please input a value for pc: 0.83
Data chosen:
L: 91.000000
C: 12.000000
V: 2.000000
td: 30.000000
pc: 0.830000
R: 1.130365


Dissipation of Charge(q) over Time(t)
q(t)
t
12
14
16
18
20
22
24
0
5
10
15
20
25

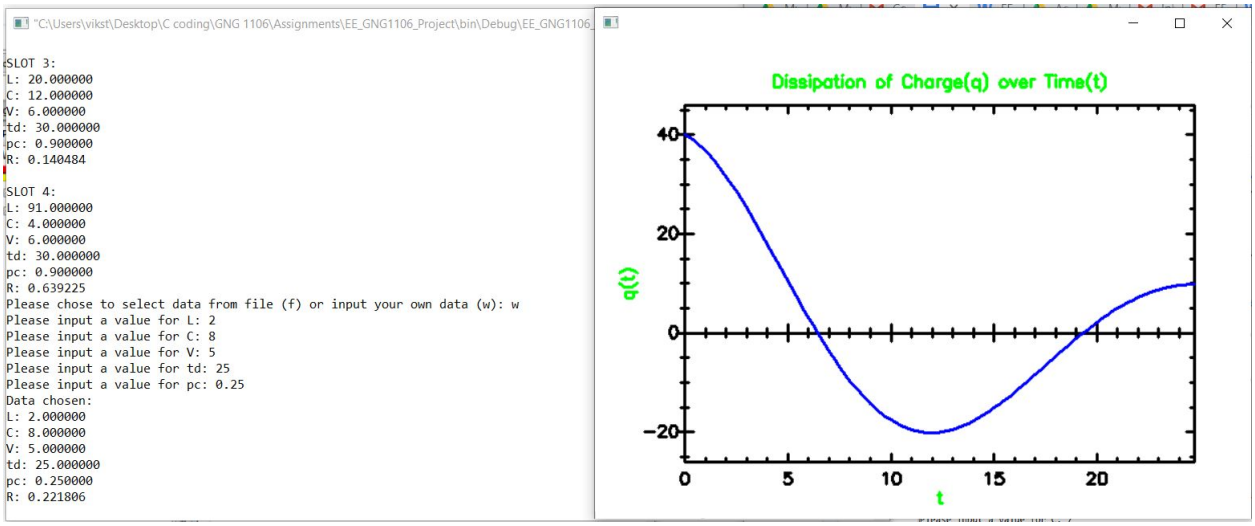
Case 6:


"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Proje
SLOT 3:
L: 20.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.140484
SLOT 4:
L: 91.000000
C: 4.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639225
Please chose to select data from file (f) or input your own d
Please input a value for L: 33
Please input a value for C: 2
Please input a value for V: 6
Please input a value for td: 30
Please input a value for pc: 0.5
Data chosen:
L: 33.000000
C: 2.000000
V: 6.000000
td: 30.000000
pc: 0.500000
R: 1.524900
Dissipation of Charge(q) over Time(t)
q(t)
10
5
0
-5
0
5
10
15
20
25
t

Case 7:


"C:\Users\vikst\Desktop\C coding\GNG 1106\Assignments\EE_GNG1106_Project\bin\Debug\EE_GNG1106_Project.exe"
SLOT 3:
L: 20.000000
C: 12.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.140484
SLOT 4:
L: 91.000000
C: 4.000000
V: 6.000000
td: 30.000000
pc: 0.900000
R: 0.639225
Please chose to select data from file (f) or input your own data (w): w
Please input a value for L: 54
Please input a value for C: 7
Please input a value for V: 2
Please input a value for td: 15
Please input a value for pc: 0.2
Data chosen:
L: 54.000000
C: 7.000000
V: 2.000000
td: 15.000000
pc: 0.200000
Values given will not yeild an R value

Case 8:



Choosing where to store data:

```
Data chosen:
L: 33.000000
C: 2.000000
V: 6.000000
td: 30.000000
pc: 0.500000
R: 1.524900
Would you like to overwrite a data slot (Y/N)? Y
Please chose a data slot (0-4): 1

Data has been saved to slot 1
```

```
SLOT 1:
L: 33.000000
C: 2.000000
V: 6.000000
td: 30.000000
pc: 0.500000
R: 1.524900
```