GNG1106 – Fundamentals of Engineering Computation
Course Project

M.E. Electrical Lab

Mathew Aoun, Elias Tarabey

Date: Friday December 9th, 2022

1 Problem Identification and Statement

Provide a brief definition of the problem statement and how the software will solve the problem. What needs to be done and why? How will the software do it? (1-2 paragraphs)

The problem that we are trying to resolve is determining the appropriate value of the inductance L for a specific inductor so that it could be conformed to the desired rate of dissipation td of the RLC circuit. This determination will be based on a given resistor (R), capacitor (C) and time dissipation values. It is also important to mention that the rate of dissipation must be depended on by an upper and lower bound limit of a desired percentage of charge (pc) for the capacitor to reach regarding its original charge.

The program that will be written to solve the problem will take in inputs of capacitance, resistance and time dissipation from the user, and these values will be processed through equations given in the prompt document (Equations 9 and 10 could be used to find the value of the inductance based off of user inputs, as well as Equation 7, which would be used to determine the charge that the capacitor would be at based on the time dissipation. This could be used to verify if the value of L would be conformed to the time dissipation given by the user.) After verifying the validity of the values of L, the results could then be graphed, and the values could be saved in a file.

Gathering of Information and Input/Output Description

Break down activities/steps to solve the problem. Provide any major equation/task/calculation at each step. The last section should provide the input for the software (user-provided information) and the output of the software (how results are presented to the user).

Parts to break down:

- Function to receive input from user.
- Function to calculate if the minimum value of L is bigger or not than the minimum threshold.
- Function to find the roots of Equation 10 by using the bisection method (used to find the values of L)
- Function to calculate the values of L based on Equation 9 and user inputs.
- Function to plot the values of L with respect to pc values calculated and stored in a 2D array.
- Function to save the values of the calculated values of L, and the graph.

1.1 Activity 1: Calculations & Background

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

1.2 Activity 2: Calculations & Background

$$p_c = e^{-\frac{Rt_d}{(2L)}}$$

1.3 Activity 3: Calculations & Background

$$g(L) = e^{-\frac{Rt_d}{(2L)}} - p_c$$

1.4 Activity 4: Calculations & Background

$$L_{min} = \frac{CR^2}{4}$$

1.5 Activity 5: Calculations & Background

$$L = \frac{-Rt_d}{2\ln(p_c)}$$

1.6 Input and Output

Input : Capacitor value (C), Resistor (R), Time Dissipation (td), Minimum and maximum capacitor percentage value (pc).

Output: Graph and table of values of the inductor (L) values with respect to pc. File of the graph and saved user inputs.

2 Test Cases and Design

2.1 Test Cases

Cases leading to errors:

- *A min pc value that is greater than max pc*
- *Negative values for C, R td, min pc and max pc*
- *If all of the inputed values are 0.*

Edge cases:

- *Min pc and max pc have the same values.*
- *Big values for R, C, td*
- *Invalid file name*

2.2 Design

2.2.1 Functions for Interacting with the User

- *Function that validates the inputs of the user*
    - *int checkinp(structvariabletype *circuitptr)*
        - *Parameters :*
            - *circuitptr : Pointer that references the structure variable where the values of R, C, td, minpc and maxpc are stored.*
        - *Return value: 1 (TRUE) or 0 (FALSE)*
        - *Logic/Algorithm:*
            - *This function implements logic statements that verify that the values of R, C, td, minpc and maxpc, and diffpc using a for loop structure to verify if the inputs are of type string or are negative.*
                - *If any of the values are negative, or are strings, the function returns the symbolic constant FALSE.*
                - *If the values are valid (they are positive and are not strings), the function returns the symbolic constant TRUE (1), which breaks out of the do while loop of the main function.*
            - *What we added:*
                - *Switch the statement so it could slowly take all the input of the user*
                - *We added case 5 that does a verifies that maxpc – minpc is positive*
                - *We created a new array in the structure to save the data of user to later on save it in a file.*
    - *void main():*
        - *Parameters: None*
        - *Return value: None*
        - *Logic/Algorithm:*
            - *The main function will be used to call the functions that will be described in this document, but it will also communicate with the*

user in order that they might enter the appropriate values when prompted.

- *The function shall also contain do while loops in order to make the user to enter a valid input. If not, the do while loop continues until the user enters a valid input.*
- *The main function shall also ask the user if they want to save their inputs in a file, and if they do, the program will call the writefile function.*
- *The main function shall also call all of the functions defined in this document.*
- *What we added:*
    - *We created a do while loop to make sure that the value of L is bigger Lmin.*
    - *At the ending of the function, we ask the user if he wants to save the data in a file*

2.2.2 Functions for Calculations

- *Function CalculateLmin*
    - *int CalculateLmin(structvariabletype * circuitptr)*
        - *Parameters:*
            - *circuitptr : Pointer that references the structure variable where the values of R, C, td, minpc and maxpc are stored.*
    - *Return value: 1 (TRUE) or 0 (FALSE)*
    - *Logic/Algorithm: This function will be used to calculate the L by using equation 7 and if we find that the value of L is bigger than the Lmin then we will tell the user to try again (do/while loop).*

- Function Calculate
    - *Double Calculate(structvariabletype * circuitptr)*
        - *Parameters:*
            - *circuitptr : Pointer that references the structure variable where the values of R, C, td, minpc and maxpc are stored.*
    - *Return Value: The value calculated*
    - *Logic/Algorithm: This function will be used to calculate the 9th and 10th equations and after the calculation, it will verify the value and the roots to make sure they are both the same if not then we will ask the user to enter new values.*
            - *What we added:*
                - *We created a while loop to calculate equation 10 (bisection root Method) to find the value of L by getting the values of pc that is incremented at each iteration and when it finds its it reset the value of xl xu and increments pc and I at each iteration and save the value of pc in the 2D array.*

See D2 Guidelines for Software Design Example.

2.2.3 Functions for Plotting

- *Plot function*
    - *void plot(structvariabletype * circuitptr, double *xPtr, double *yPtr)*
        - *Parameters:*
            - *n: number of points in the arrays*
            - **xPtr: pointer to x values (horizontal axis).*
            - **yPtr: pointer to y values (vertical axis).*
        - *Return value: None*
        - *Description: Initialises the plot. The following values in the referenced structure are used to set up the plot: x[0], x[n-1] also assume that x values are sequential Miny, Maxy - vertical axis range. Also, sets up a white background and black foreground colours. Then plots the curve accessed using *xPtr and *yPtr.*
        - *Logic/Algorithme: This function is used to plot the graph by adding the x and y axes, changing the colour of the text labels, changing the colour of the plotted curve and thickening the plotted curve etc. Also, in this function, we will be calling 2 extra functions (getMin and getMax) and they will be gather the maximum and minimum values found in the array.*

- *Function plotFunction*
    - *void plotFunc(structvariabletype * circuitptr)*
        - *Parameters:*
            - *Begin: begin on the intervals (x values) to plot*
            - *End: end of interval (x values) to plot*
            - *Description: Plot the function on the interval between begin and the end. Plots an x at the roots*
            - *Logic/Algorithm: This function is used to first plot the beginning and the end of the x-axes but also, it calls the function getcalculation and returns the value of f(x) will plot the value of f(x) on the graph*
            - *We added: it only puts the value of pc and L in 2 different arrays so we can put it back in a function when we call it*
- *Function ReadFile*
    - *double ReadFile (structvariabletype * circuitptr)*
        - *Parameters:*
            - ** structvariabletype * circuitptr*
        - *Return Value: True or False*
        - *Description: This function is used to read a file*
        - *Logic/Algorithm: The functions start by asking the user if they want to read from a file. If the users say yes, then they will have to enter the file name. If not, then the programming code will asking them to put the input needed.*

- *Function WriteFile*
    - *double WriteFile (structvariabletype * circuitptr)*
        - *Parameters :*
            - ** structvariabletype * circuitptr*

        - Return Value: None
        - Description: This function is used to write in the file
        - Logic/Algorithm: This function is used at the end of the code when you are done getting all your data the code is gonna ask you to if you wanna save it in a file and if yes they will ask you to write the name of the code so later on if you would like to return those resulats they will be under a file name.

See D2 Guidelines for Software Design Example.

2.2.4 Functions for Files

- A function that writes in a new or existences file
    - void writefile(structvariabletype *eleccircuitpointer)
        - Parameters: Reference (pointer) to structure variable containing the values of R, C, td, minpc and maxpc.
        - Return Value: Returns nothing (void)
        - Logic/Expressions:
            - A for loop is executed and an index variable is declared before the execution of the loop, and this for loop shall write in a created file the values of R, C, td, minpc and maxpc in this order in a file on the same line.
            - The function shall also write to an existing if there is already at least one set of valid values that are present within the file.
            - An array shall be declared in the structure variable that will contain all of the values for C, R, td, minpc and maxpc, and this permits to index and loop through all of the values within the array.
            - If a file does not exist, fprintf shall be called in order to create an ASCII file so that the user could save their inputs in one line of the file.

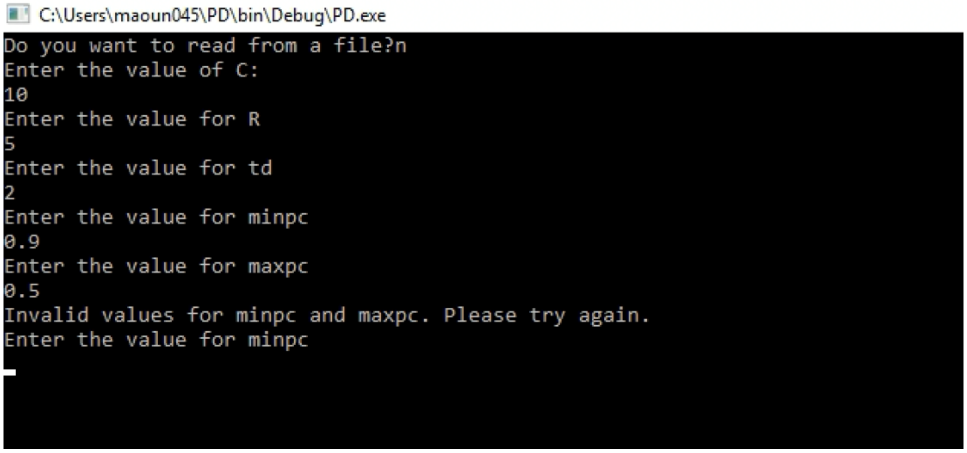
- Function to read the file
    - int readfile(structvariablename *eleccircuitpointer, int linenumber)
        - Parameters:
            - Structure variable pointer that references the structure variable that contains the values of R, C, td, minpc and maxpc. An array that contains these values is also present within the structure type.
            - An integer that specifies the line number to read.
        - Return type: Returns TRUE (1) or FALSE (0) if the values of the file are successfully put into the variables of the structure variable.
        - Logic/Expressions:
            - This function shall read in values of a file based on the specified line that the user inputted in the main program.
            - A logic expression shall be implemented to verify if the line indicated by the user exists or not. If not, an error message shall be displayed to the user.

- *A for loop expression shall be used to assign each consequent value to the appropriate variable of the structure variable and the function shall return TRUE if the values were stored, and FALSE if they were not.*

3 Test Cases and Design

3.1 Test Cases

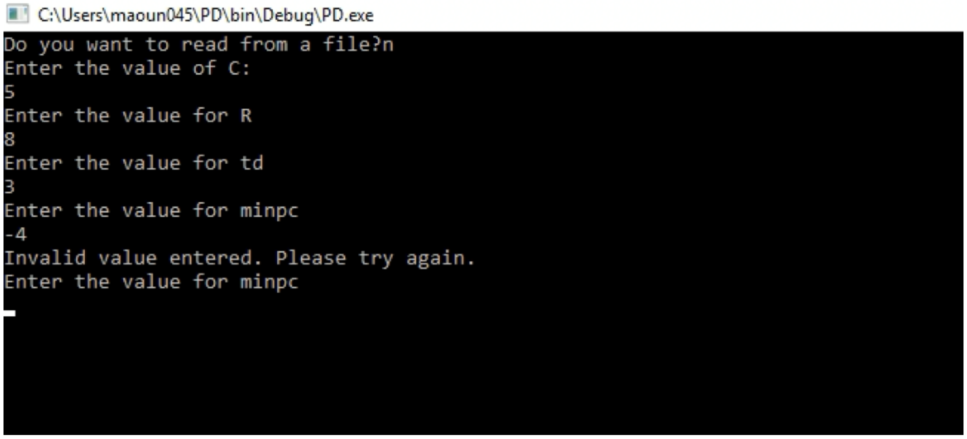
Cases leading to errors:
A min pc value that is greater than max pc



C:\Users\maoun045\PD\bin\Debug\PD.exe

```
Do you want to read from a file?n
Enter the value of C:
10
Enter the value for R
5
Enter the value for td
2
Enter the value for minpc
0.9
Enter the value for maxpc
0.5
Invalid values for minpc and maxpc. Please try again.
Enter the value for minpc
```

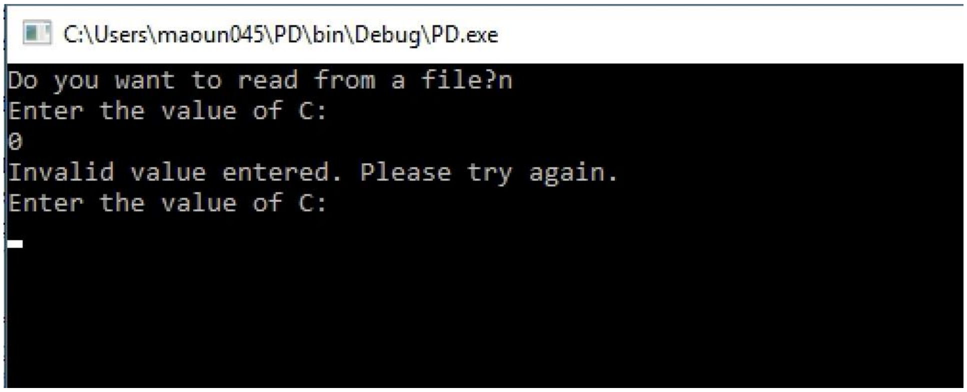
Negative values for C, R td, min pc and max pc



C:\Users\maoun045\PD\bin\Debug\PD.exe

```
Do you want to read from a file?n
Enter the value of C:
5
Enter the value for R
8
Enter the value for td
3
Enter the value for minpc
-4
Invalid value entered. Please try again.
Enter the value for minpc
```

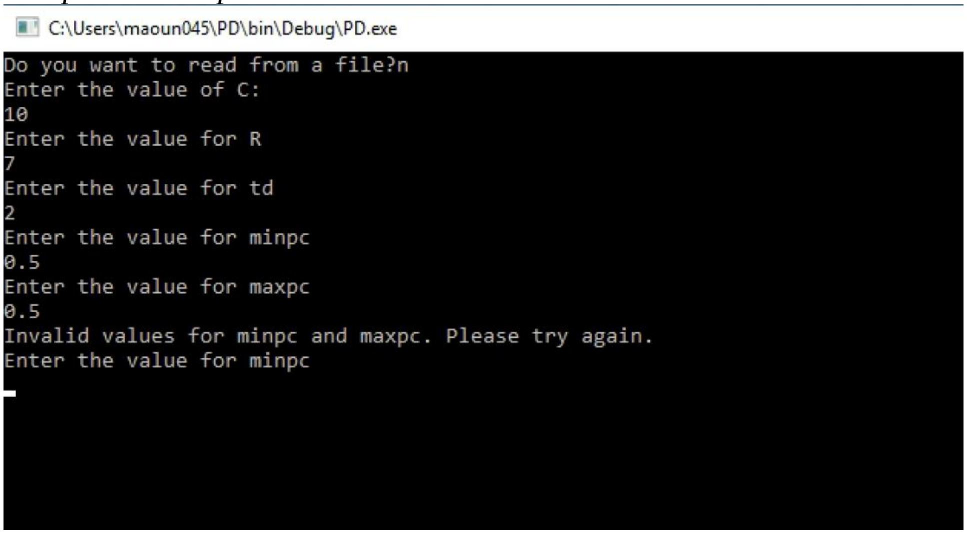
If all of the inputed values are 0.



```
Do you want to read from a file?n
Enter the value of C:
0
Invalid value entered. Please try again.
Enter the value of C:
```

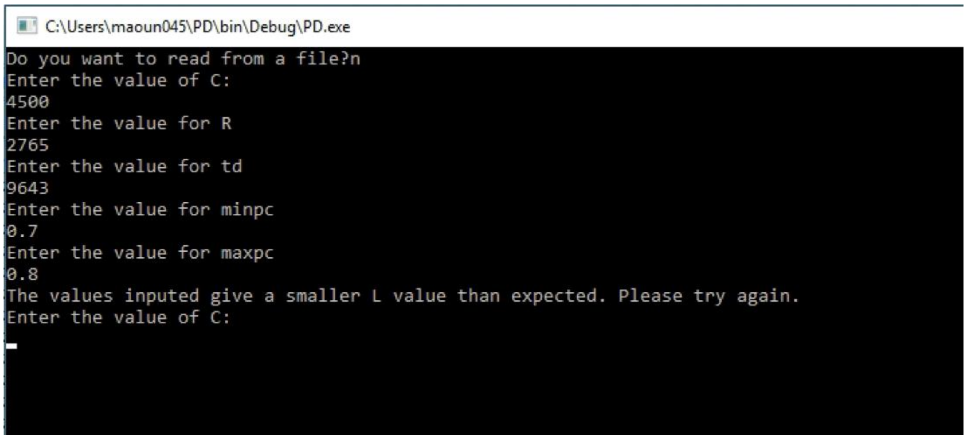
Edge cases:

Min pc and max pc have the same values.



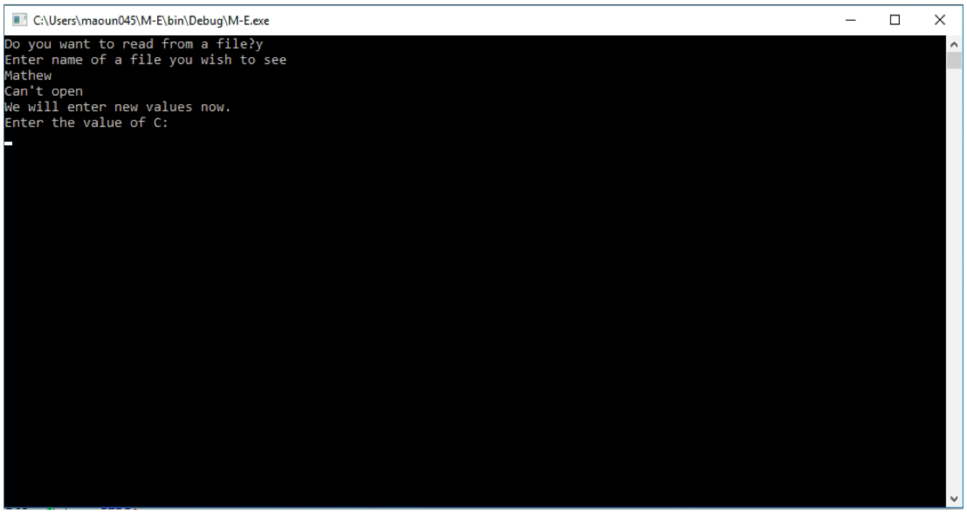
```
Do you want to read from a file?n
Enter the value of C:
10
Enter the value for R
7
Enter the value for td
2
Enter the value for minpc
0.5
Enter the value for maxpc
0.5
Invalid values for minpc and maxpc. Please try again.
Enter the value for minpc
```

Big values for R, C, td



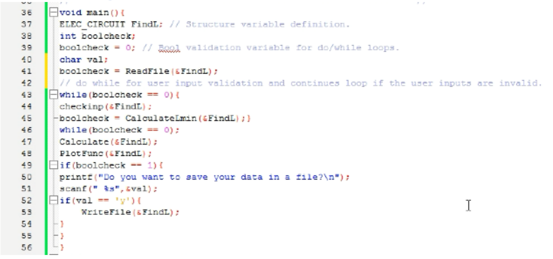
```
Do you want to read from a file?n
Enter the value of C:
4500
Enter the value for R
2765
Enter the value for td
9643
Enter the value for minpc
0.7
Enter the value for maxpc
0.8
The values inputed give a smaller L value than expected. Please try again.
Enter the value of C:
```

Invalid file name



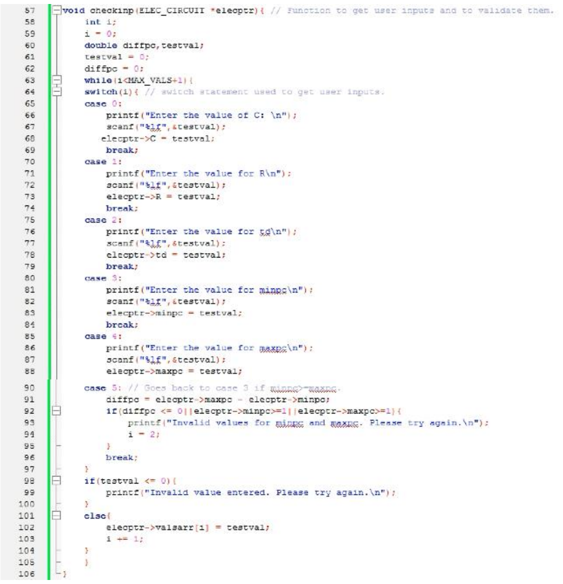
```
Do you want to read from a file?y
Enter name of a file you wish to see
Mathew
Can't open
We will enter new values now.
Enter the value of C:
_
```

3.2 Design



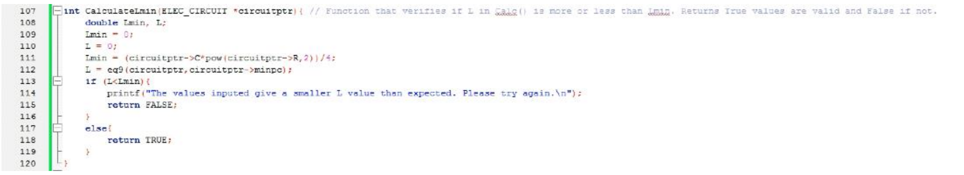
```c
void main(){
ELEC_CIRCUIT FindL; // Structure variable definition.
int boolcheck;
boolcheck = 0; // Bool validation variable for do/while loops.
char val;
boolcheck = ReadFile(&FindL);
// do while for user input validation and continues loop if the user inputs are invalid.
while(boolcheck == 0){
checkinp(&FindL);
boolcheck = CalculateLmin(&FindL);}
while(boolcheck == 0);
Calculate(&FindL);
PlotFunc(&FindL);
if(boolcheck == 1){
printf("Do you want to save your data in a file?\n");
scanf(" %s",&val);
if(val == 'y'){
    WriteFile(&FindL);
}
}
}
```

3.2.1 Functions for Interacting with the User

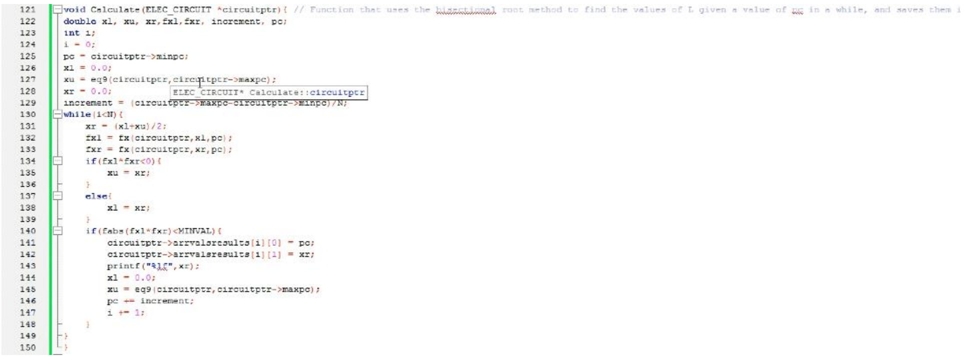


```c
void checkinp(ELEC_CIRCUIT *elecptr){ // Function to get user inputs and to validate them.
    int i;
    i = 0;
    double diffpc,testval;
    testval = 0;
    diffpc = 0;
    while(i<MAX_VALS+1){
    switch(i){ // switch statement used to get user inputs.
    case 0:
        printf("Enter the value of C: \n");
        scanf("%lf",&testval);
      elecptr->C = testval;
        break;
    case 1:
        printf("Enter the value for R\n");
        scanf("%lf",&testval);
        elecptr->R = testval;
        break;
    case 2:
        printf("Enter the value for td\n");
        scanf("%lf",&testval);
        elecptr->td = testval;
        break;
    case 3:
        printf("Enter the value for minpc\n");
        scanf("%lf",&testval);
        elecptr->minpc = testval;
        break;
    case 4:
        printf("Enter the value for maxpc\n");
        scanf("%lf",&testval);
        elecptr->maxpc = testval;

    case 5: // Goes back to case 3 if minpc>=maxpc.
        diffpc = elecptr->maxpc - elecptr->minpc;
        if(diffpc <= 0||elecptr->minpc>=1||elecptr->maxpc>=1){
            printf("Invalid values for minpc and maxpc. Please try again.\n");
            i = 2;
        }
        break;
    }
    if(testval <= 0){
        printf("Invalid value entered. Please try again.\n");
    }
    else{
        elecptr->valsarr[i] = testval;
        i += 1;
    }
    }
}
```

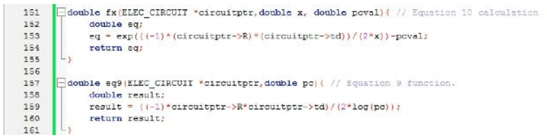
3.2.2 Functions for Calculations



```
int CalculateLmin(ELEC_CIRCUIT *circuitptr){ // Function that verifies if L in Calc() is more or less than Lmin. Returns True values are valid and False if not.
    double Lmin, L;
    Lmin = 0;
    L = 0;
    Lmin = (circuitptr->C*pow(circuitptr->R,2))/4;
    L = eq9(circuitptr,circuitptr->minpc);
    if (L<Lmin){
        printf("The values inputed give a smaller L value then expected. Please try again.\n");
        return FALSE;
    }
    else{
        return TRUE;
    }
}
```

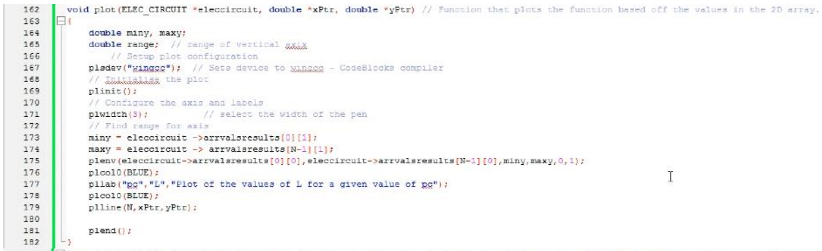


```
void Calculate(ELEC_CIRCUIT *circuitptr){ // Function that uses the bisectional root method to find the values of L given a value of pc in a while, and saves them i
double xl, xu, xr,fxl,fxr, increment, pc;
int i;
i = 0;
pc = circuitptr->minpc;
xl = 0.0;
xu = eq9(circuitptr,circuitptr->maxpc);
xr = 0.0;         ELEC_CIRCUIT* Calculate::circuitptr
increment = (circuitptr->maxpc-circuitptr->minpc)/N;
while(i<N){
    xr = (xl+xu)/2;
    fxl = fx(circuitptr,xl,pc);
    fxr = fx(circuitptr,xr,pc);
    if(fxl*fxr<0){
        xu = xr;
    }
    else{
        xl = xr;
    }
    if(fabs(fxl*fxr)<MINVAL){
        circuitptr->arrvalsresults[i][0] = pc;
        circuitptr->arrvalsresults[i][1] = xr;
        printf("%lf",xr);
        xl = 0.0;
        xu = eq9(circuitptr,circuitptr->maxpc);
        pc += increment;
        i += 1;
    }
}
}
```

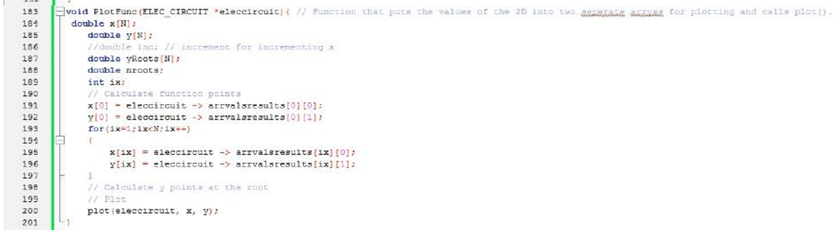


```
double fx(ELEC_CIRCUIT *circuitptr,double x, double pcval){ // Equation 10 calculation
    double eq;
    eq = exp(((-1)*(circuitptr->R)*(circuitptr->td))/(2*x))-pcval;
    return eq;
}

double eq9(ELEC_CIRCUIT *circuitptr,double pc){ // Equation 9 function.
    double result;
    result = ((-1)*circuitptr->R*circuitptr->td)/(2*log(pc));
    return result;
}
```

3.2.3 Functions for Plotting

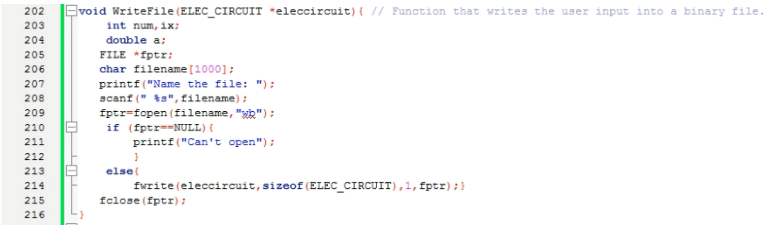


```
 void plot(ELEC_CIRCUIT *eleccircuit, double *xPtr, double *yPtr) // Function that plots the function based off the values in the 2D array.
{
     double miny, maxy;
     double range;  // range of vertical axix
         // Setup plot configuration
     plsdev("wingcc");  // Sets device to wingcc - CodeBlocks compiler
     // Initialise the plot
     plinit();
     // Configure the axis and labels
     plwidth(3);          // select the width of the pen
     // Find range for axis
     miny = eleccircuit ->arrvalsresults[0][1];
     maxy = eleccircuit -> arrvalsresults[N-1][1];
     plenv(eleccircuit->arrvalsresults[0][0],eleccircuit->arrvalsresults[N-1][0],miny,maxy,0,1);
     plcol0(BLUE);
     pllab("pc","L","Plot of the values of L for a given value of pc");
     plcol0(BLUE);
     plline(N,xPtr,yPtr);

     plend();
}
```

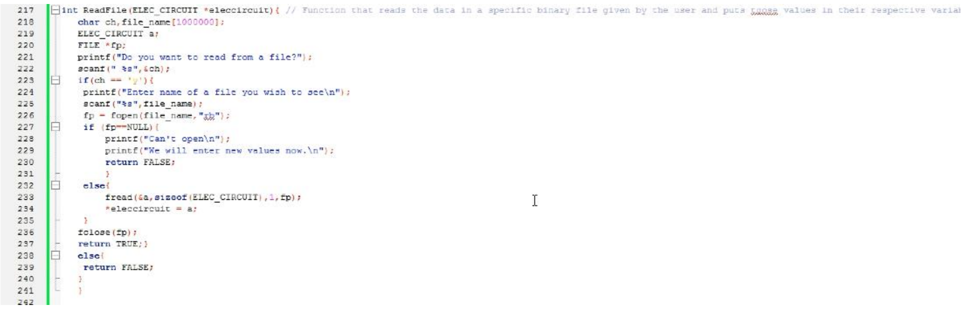


```
void PlotFunc(ELEC_CIRCUIT *eleccircuit){ // Function that puts the values of the 2D into two seperate arryas for plotting and calls plot().
 double x[N];
    double y[N];
    //double inc; // increment for incrementing x
    double yRoots[N];
    double nroots;
    int ix;
    // Calculate function points
    x[0] = eleccircuit -> arrvalsresults[0][0];
    y[0] = eleccircuit -> arrvalsresults[0][1];
    for(ix=1;ix<N;ix++)
    {
        x[ix] = eleccircuit -> arrvalsresults[ix][0];
        y[ix] = eleccircuit -> arrvalsresults[ix][1];
    }
    // Calculate y points at the root
    // Plot
    plot(eleccircuit, x, y);
}
```

3.2.4 Functions for Files



```c
void WriteFile(ELEC_CIRCUIT *eleccircuit){ // Function that writes the user input into a binary file.
     int num,ix;
     double a;
    FILE *fptr;
    char filename[1000];
    printf("Name the file: ");
    scanf(" %s",filename);
    fptr=fopen(filename,"wb");
     if (fptr==NULL){
         printf("Can't open");
         }
     else{
         fwrite(eleccircuit,sizeof(ELEC_CIRCUIT),1,fptr);}
    fclose(fptr);
}
```



```c
int ReadFile(ELEC_CIRCUIT *eleccircuit){ // Function that reads the data in a specific binary file given by the user and puts tqose values in their respective variab
    char ch,file_name[1000000];
    ELEC_CIRCUIT a;
    FILE *fp;
    printf("Do you want to read from a file?");
    scanf(" %s",&ch);
    if(ch == 'y'){
     printf("Enter name of a file you wish to see\n");
     scanf("%s",file_name);
     fp = fopen(file_name,"rb");
     if (fp==NULL){
         printf("Can't open\n");
         printf("We will enter new values now.\n");
         return FALSE;
         }
     else{
         fread(&a,sizeof(ELEC_CIRCUIT),1,fp);
         *eleccircuit = a;
     }
    fclose(fp);
    return TRUE;}
    else{
     return FALSE;
    }
    }
```

4 Implementation

See the C program in file project.c