CSE1PES: PROGRAMMING FOR ENGINEERS AND SCIENTISTS

ASSIGNMENT 2 (10%)

TOOLS

- Unix server through Putty for compiling and testing
- Notepad ++ for writing solutions
- LMS for submission
- Microsoft word (or similar) for writing written answers

SUBMISSION

The task is split into two submissions:

- **The code**
  - **Submitted as a .c file in the normal assignment submission portal. (Do not submit the executable.)**
- **The written portion**
  - **Submitted as a PDF to the turnitin assignment portal.**

Each has a separate submission portal, ensure to submit both.

There is NO late submissions for this assignment. No assignment will be accepted after the due date, without a legitimate extension granted by the lecturer before the due date.

ACADEMIC INTEGRITY

Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. For individual assignments, plagiarism includes the case where two or more students work collaboratively on the assignment. The School of Engineering and Mathematics treats plagiarism very seriously. When it is detected, penalties are strictly imposed.
http://www.latrobe.edu.au/students/academic-integrity

IMPEDANCE, VOLTAGE AND CURRENT CALCULATOR FOR SIMPLE AC CIRCUIT

BACKGROUND

An Alternating Current (AC) circuit is one in which the output of the voltage source changes with time. When we add components to the circuit, the voltage and current will be different at different points in the circuit. The components we will be looking at are resistors, capacitors and inductors. Further information can be found here: https://en.wikipedia.org/wiki/Alternating_current, https://en.wikipedia.org/wiki/Inductor, https://en.wikipedia.org/wiki/Resistor, https://en.wikipedia.org/wiki/Capacitor.

Impedance is the measure of the resistance and reactance of a component or circuit. Resistance is a measure of friction against the motion of electrons, whereas reactance is the inertia against the motion of electrons. Impedance introduces a phase shift between the voltage and current through a component, and hence is expressed as a complex number. **Do not worry too much if this doesn't make sense, you can just think of this as complex resistance.** For further information, please read https://en.wikipedia.org/wiki/Electrical_impedance.

To calculate the impedance, voltage and current for each component in the circuit, we can use ohms law. The equation for ohms law is as follows: **(Note we are dealing with complex numbers, however we will develop all the necessary functions you will need in lab 5)**

$$V = I * Z \,, I = \frac{V}{Z} \,, Z = \frac{V}{I}$$

Where V, I and Z represent voltage, current and impedance respectively, and are measured in volts, amps and ohms respectively.

For series circuits the current is the same through all components.
For parallel circuits the voltage is the same through all components.

The impedance calculation depends on the component. Details are as follows:

Resistors: The impedance for a resistor can be found as:

$Z = R$ where R is the resistance measured in ohms.

Capacitors: The impedance for a capacitor can be found as:

$X_c = \frac{1}{2*\pi*f*C}$ where C is the capacitance measured in Farads and f is the frequency measured in hertz

$Z = 0 - j * X_c$ where X_c is the reactance and j is the imaginary unit.

Inductors: The impedance for an inductor can be found as:

$X_L = 2 * \pi * f * L$ where L is the inductance measured in Henrys

$Z = 0 + j * X_L$ where X_L is the reactance and j is the imaginary unit.

Components will either be arranged in parallel or series. Depending on this, the total impedance calculation will vary.

The total impedance between two components in series is:

$$Z_T = Z_1 + Z_2$$

If we have three components in series, then the total impedance is:

$$Z_{T2} = Z_T + Z_3$$

Hence, generalizing this we get:

$$Z_T = \sum_{k=1}^{n} Z_k$$

The total impedance between two components in parallel is:

$$Z_T = \left(\frac{1}{Z_1} + \frac{1}{Z_2}\right)^{-1}$$

Which we can then extend to calculate for three components by:

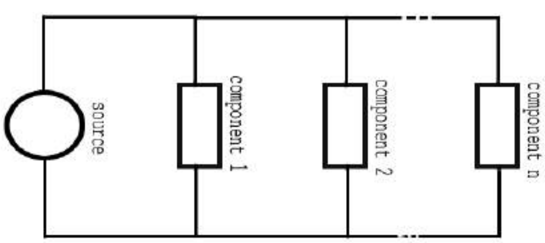
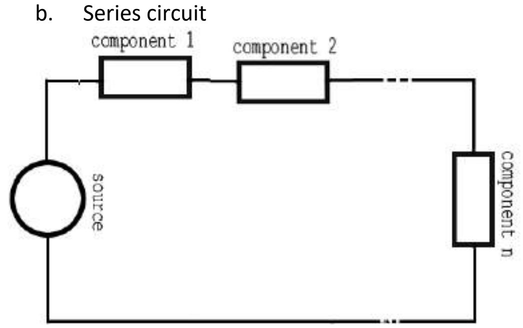
$$Z_{T2} = \left(\frac{1}{Z_T} + \frac{1}{Z_3}\right)^{-1}$$

Hence, generalizing this we get:

$$Z_T = \left(\sum_{k=1}^{n} \frac{1}{Z_k}\right)^{-1}$$

PROBLEM

You are to create a program that allows a user to view the voltage, current and impedance for each component and the total in either the parallel circuit (a) or series circuit (b). The program will ask the user for the circuit type, the source voltage, the frequency, the number of components, and the values of each component. The two circuits are shown below:

a. Parallel circuit	b. Series circuit
	

Worked Example:

Given a parallel circuit (a), with a source voltage of 12Vpp, a frequency of 60Hz, with 3 components in **parallel**.
The components are a 40 ohm resistor, a 0.0003 farad capacitor and a 0.2 henry inductor.

Firstly we calculate the impedance of each component:

$$Z_R(impedance\ of\ resistor) = 40\ ohms$$
$$Z_C(impedance\ of\ capacitor) = 0 - \frac{1}{2*\pi*60*0.0003} * j = 0 - 8.842j\ ohms$$
$$Z_L(impedance\ of\ inductor) = 0 + 2*\pi*60*0.2*j = 0 + 75.4j\ ohms$$

We now calculate the total impedance. As we are using a **parallel circuit** we use the formula as above.

We calculate the summation first as follows:

$$\sum_{k=1}^{n}\frac{1}{Z_k} = \left(\frac{1}{Z_R}+\frac{1}{Z_C}+\frac{1}{Z_L}\right)$$
$$= \frac{1}{40}+\frac{1}{-8.842j}+\frac{1}{75.4j} = \frac{40}{40^2}-\frac{8.842}{8.842^2}\mathrm{j}+\frac{75.4}{75.4^2}\mathrm{j}$$
$$\approx 0.025 + 0.1131\mathrm{j} - 0.0133\mathrm{j}$$
$$\approx 0.025 + 0.0998\mathrm{j}$$

Then find the reciprocal to get the total impedance:

$$Z_T = \left(\sum_{k=1}^{n}\frac{1}{Z_k}\right)^{-1}$$
$$\approx \frac{1}{0.025+0.0998\mathrm{j}}$$
$$\approx \frac{0.025}{0.025^2+0.0998^2} + j\frac{-0.0998}{0.025^2+0.0998^2}$$
$$\approx 2.36 - j9.429$$

We then calculate the total current using Ohms law

$$I = \frac{V}{Z} = \frac{12}{2.36-9.429j} = 0.3 + 1.198j\ Amps$$

We then know that voltage is the same throughout the circuit because all the components are in parallel.

So we only need to calculate the current for each component:

$$I_R = \frac{12}{40} = 0.3\ Amps$$
$$I_C = \frac{12}{-8.842j} = 1.357j\ Amps$$
$$I_L = \frac{12}{75.4j} = -0.159j\ Amps$$

Hence we can now generate the table:

COMPONENTS	R	C	L	TOTAL
Z (Ohms)	40+0j	0-8.842j	0+75.4j	$2.36 - 9.425j$
I (Amps)	0.3+0j	0+1.357j	0-0.159j	0.3+1.198j
V (Volts)	12+0j	12+0j	12+0j	12+0j

FUNCTION PROTOTYPES AND INCLUDES

1. You must include the stdio.h and stdlib libraries
2. The following function prototypes are to be used:

```
void calculate_impedance(double* result_table[6], int components[], double values[], int n, int freq, int choice);
void print_results(double* result_table[6], int n, int components[]);

void complex_addition(double in1_real, double in1_imag, double in2_real, double in2_imag, double* out3_real, double* out3_imag);
void complex_subtraction(double in1_real, double in1_imag, double in2_real, double in2_imag, double* out3_real, double* out3_imag);
void complex_multiplication(double in1_real, double in1_imag, double in2_real, double in2_imag, double* out3_real, double* out3_imag);
void complex_division(double in1_real, double in1_imag, double in2_real, double in2_imag, double* out3_real, double* out3_imag);
```

   a. The first two are described below
   b. The last four will be developed in lab5, you need to alter the types from floats to doubles
   c. You **can** change the naming
   d. You **cannot** change the types

FUNCTION - MAIN

1. The program prints to the screen the student number, student name and the assignment number. This is enclosed by double asterisks. Ensure you follow the below format.

```
**3777777 John Smith Assignment 2**
```

2. The program asks the user which circuit they want to evaluate, the frequency in hertz, the source voltage and the number of components. And stores the results appropriately. **All of these values are positive integers**. When asking for the number of components, the type of circuit should be specified as shown below.
   a. You can assume the user enters valid values for frequency, voltage and number of components.
   b. You must handle the case where the user enters an incorrect value for the type of circuit and give a message to the user. **(No example given, a single line for 'invalid message' to user ending with a newline. The message is up to you, but must be appropriate)**

```
Do you want to create a 1.Parallel or 2.Series circuit evaluation? 1
Enter the frequency in Hertz of the source: 60
Enter the source voltage: 12
How many components are in parallel: 3
```

3. The program creates an array of **pointers to doubles** *(hint: use [ ]),* the array is of length 6
   a. Then using a **for loop** dynamically allocate memory (malloc) for each of the pointers an array of size **sizeof(double) *(*number of components* + 1)**
   b. This is the result table
   c. The program stores the source voltage in the array as shown below

```
result_table[4][number_of_components]=source_voltage;
result_table[5][number_of_components]=0L;
```

	0	*1*	*…*	*number of components -1*	*number of components*
0	Impedance component 1 – Real part	Impedance component 2 – Real part	…	Impedance component n – Real part	Total Impedance - Real part
1	Impedance component 1 – Imaginary part	Impedance component 2 – Imaginary part	…	Impedance component n – Imaginary part	Total Impedance - Imaginary part
2	Current component 1 – Real part	Current component 2 – Real part	…	Current component n – Real part	Total Current – Real part
3	Current component 1 – Imaginary part	Current component 2 – Imaginary part	…	Current component n – Imaginary part	Total Current – Imaginary part
4	Voltage component 1 – Real part	Voltage component 2 – Real part	…	Voltage component n – Real part	Total Votage/Source Voltage – Real part
5	Voltage component 1 - Imaginary part	Voltage component 2 – Imaginary part	…	Voltage component n – Imaginary part	Total Votage/Source Voltage – Imaginary part

4. The program creates a pointer to ints and dynamically allocates memory for an array **of length** *number of components*. This will store the component types.
5. The program creates a pointer to doubles and dynamically allocates memory for an array **of length** *number of components.* This will store the size of the components.
6. For each component, the program asks the user to enter the type of component and the value of that component. When asking for the size, the correct units must be shown (Ohms, Farads and Henrys).
    a. **<u>The storing of the component types and values must be done using pointer arithmetic in this section.</u>** (Refer to lab 6 and additional material)
    b. You can assume the user enters valid values components (1,2 or 3) and their values (positive values)

```
Enter the component type for each component where 1 is Resistor, 2 is Capacitor and 3 is Inductor
Component 1: 1
Enter the size of the component in Ohms: 40
Component 2: 2
Enter the size of the component in Farads: 0.0003
Component 3: 3
Enter the size of the component in Henrys: 0.2
```

7. Calculate the impedance values by calling the calculate_impedance function. (Described in section below)
8. Calculate the total current values for both the real and imaginary parts and store appropriately (Ohms law I=V/Z)
9. Calculate the voltages and currents of all components, and store appropriately. *(Hint. Refer to background information – Ohms Law. Parallel and series will have different equations)*
10. Call the print_results function to print out the results table (Described in section below)
11. End the program appropriately.

FUNCTION - CALCULATE_IMPEDANCE

1. Calculate the **impedance** for both real and imaginary part for each component using a for loop with an if, else-if, else statement in the loop. And store the results in the results_table (Hint. Refer to background information – Impedance. Resistors, Capacitors and Inductors have different equations)
2. Calculate the total impedance for both real and Imaginary parts. And store the results in the results_table (Refer to additional material if new to summation)
    a. Hint: (One possible implementation)
        i. Create temporary variables to store the current sum (for both complex and real parts) and initialize to 0.
        ii. Create variables to store the total impedance (both for complex and real parts).
        iii. Then if the circuit is series:
            1. Iteratively add to these for each impedance values.
            2. The sum is the total impedance
        iv. else if the circuit is parallel:
            1. Create a set of temporary variables for the result of the division
            2. Iterate through the component values starting from the first component:
                i. Divide 1+0j by the components impedance value and store the result appropriately, then add this result to the current sum.
            3. Divide 1+0j by the total sum to give the total impedance.
        v. Store the total impedance values in the appropriate place in the result table.

FUNCTION - PRINT_RESULTS

1. Print the result table as shown below:
    a. The first column describes the rows with the text "COMPONENTS","Z (Ohms)","I (Amps)","V (Volts)".

b. The first row prints the components as ‘R’ for resistor, ‘C’ for capacitor and ‘I’ for inductor.

c. The second row prints the impedance, the third the current and the fourth the voltage

   i. The real part is printed in brackets in exponential format with sign and 3 decimal places. Then the ‘+’ character. Then the imaginary part is printed in brackets in exponential format with sign and 3 decimal places. Then the ‘j’ character.

```
COMPONENTS |              R              |              C              |              I              |            TOTAL            |
Z ( Ohms)  |(+4.000e+00)+(+0.000e+00)j|(+0.000e+00)+(-8.842e+00)j|(+0.000e+00)+(+7.540e+00)j|(+2.360e+00)+(-9.426e+00)j|
I ( Amps)  |(+3.000e-00)+(+0.000e+00)j|(+0.000e+00)+(+1.357e+00)j|(+0.000e+00)+(-1.592e-00)j|(+3.000e-00)+(+1.198e+00)j|
V (Volts)  |(+1.200e+00)+(+0.000e+00)j|(+1.200e+00)+(+0.000e+00)j|(+1.200e+00)+(+0.000e+00)j|(+1.200e+00)+(+0.000e+00)j|
```

CONSTRAINTS

- Pi needs to be to 11 decimal places in your program.
- Formatting needs to be exactly as specified. (Your program will be tested against automated test cases)
- Text to user needs to be easily understandable. You can change the text but the same inputs must be used.
- The program must print your student number, name and the assignment number as specified.
- Types should be used appropriately.
- You must use comments to explain significant lines of code.
- You must use comments to explain how to use the functions and solution

BONUS +5% CAPPED AT 100%

- *Given on LMS, assignment tab*

HINTS

- The code should be around 150 - 200 lines of code without the bonus. If your code is significantly larger you may want to reconsider your approach.

SUBMISSION

Part 1. Solve the problem by implementing a program using C code.

Part 2. Produce a report of 500 +/- 150 words explaining:

- 1. Explain how you implemented step 2.b in “FUNCTION – MAIN”, why you implemented in this way and explain why it is important to validate inputs.
- 2. Explain how step 3 in “FUNCTION – MAIN” is implemented, why it is implemented in this way and how memory is allocated in this step.
- 3. Explain what happens when the ‘calculate_impedence’ function is called, how the variables are passed and how they are returned. Include a diagram (such as box and pointer diagram or similar) to help in the explanation.

EXAMPLES (WITHOUT BONUS)

- *Given on LMS, assignment tab*

RUBRIC

Part 1 – 74 points			
Is the student's number, student name and assignment number printed?	ACADEMIC INTEGRITY If the names are different	0 Incorrectly formatted	1 Correctly formatted
Does the program compile?	0 Does not compile.	4 Compiles with warnings.	8 Compiles appropriately.
Is the logic of the code appropriate?	0 The code logic does not match the problem.	2 Some small errors in interpreting the problem.	4 The logic is suitable for the problem.
Are the inputs and outputs understandable?	0 Does not accept inputs correctly or does not output data appropriately.	2 Some errors in outputs.	4 Inputs and outputs are handled appropriately.
Suitable comments are used to explain particular blocks or lines of code.	0 No comments	4 Reasonable comments used throughout.	8 Excellent descriptions and commenting.
Is the formatting of the table correct?	0 The tables do not use any formatting.	4 The tables are formatted but incorrectly.	6 Correct format specifiers, spacing and new lines are used
Is the impedance calculated correctly	0 The impedance is not calculated correctly	4 The impedance for each component is calculated correctly, but not the total for both series and parallel.	9 Total impedance and impedance for each component is calculated correctly for both series and parrallel.
Is the current calculated correctly	0 The current is not calculated correctly.	4 Total current or current of each component is calculated correctly for both series and parallel.	6 Total current and current of each component is calculated correctly for both series and parrallel.
Is the voltage calculated correctly	0 The current is not calculated correctly.	4 Voltage of each component is calculated correctly for both series and parallel.	6 Total voltage and voltage of each component is calculated correctly for both series and parrallel.
Is malloc used appropriately	0 Malloc is not used	2 Malloc is used incorrectly	4 Malloc is used appropriately
Is pointer arithmetic used appropriately	0 Pointer arithmetic is not used	2 Pointer arithmetic is used incorrectly	4 Pointer arithmetic is used appropriately
Are pointers used appropriately	0 Pointers are not used	2 Pointers are used incorrectly	4 Pointers are used appropriately
Are functions used appropriately	0 Functions are not used	2 Functions are used incorrectly	4 Functions are used appropriately
Is the code correctly indented?	0 The code is not indented.	1 Some errors in indenting.	2 Code is indented appropriately.

Are variables named appropriately?	0 Variables are named poorly.	1 Variables are named suitably for the code.	2 Variables are named suitably for the code and the domain.
Are appropriate variable types used where appropriate?	0 All variables are declared as the largest types	1 Some types are not suitable.	2 All variable types are suitable.
Part 2 – 26 points			
Is the student's number and student name on the report?	ACADEMIC INTEGRITY If the names are different		0 Included
Question 1	0 Poor explanation.	4 Reasonable explanation.	6 Excellent explanation.
Question 2	0 Poor explanation.	4 Reasonable explanation.	8 Excellent explanation.
Question 3	0 Poor explanation.	4 Reasonable explanation.	8 Excellent explanation.
Does the explanation use appropriate grammar and spelling?	0 Poor grammar or many spelling mistakes	2 Some errors in grammar or spelling.	4 Reasonable grammar and spelling.