STEP 1: Problem Identification and Statement

Problem Statement:
Calculate the total resistance in ohms for two different types of connections: series and parallel. Identify a connection type which can be either series or parallel, the number of resistors, and values of each resistor.

STEP 2: Gathering Information and Input/Output diagram

We want to find the total resistance in ohms. In order to do so, there is some necessary information that we need to identify first. Those are listed below as:

- Connection type: Series or Parallel

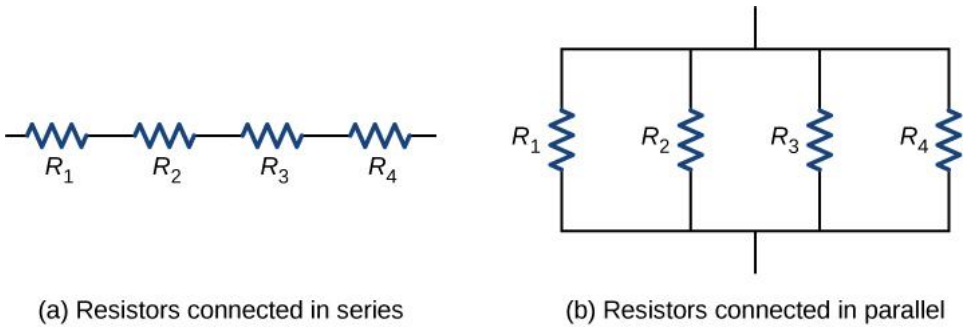


Figure 1: Series Connection (Left) and Parallel Connection (Right) of 4 resistors

- Formula for each connection type
  Series Connection:

$$R_{tot} = R_1 + R_2 + R_3 + \cdots + R_N,$$

Figure 2: Formula for finding total resistance in series connection

Parallel Connection:

$$\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \cdots + \frac{1}{R_N}.$$

Figure 3: Formula for finding total resistance in Parallel connection

- The number of resistors (N)
- Values of each resistor (R1, R2, etc…)

The below diagram shows an input/output (I/O) diagram, where the black box represents the computer program.

The inputs are: connection type, the number of resistors (N), and values of each resistor (R1, R2, etc…).
The outputs are: total resistance in ohms.

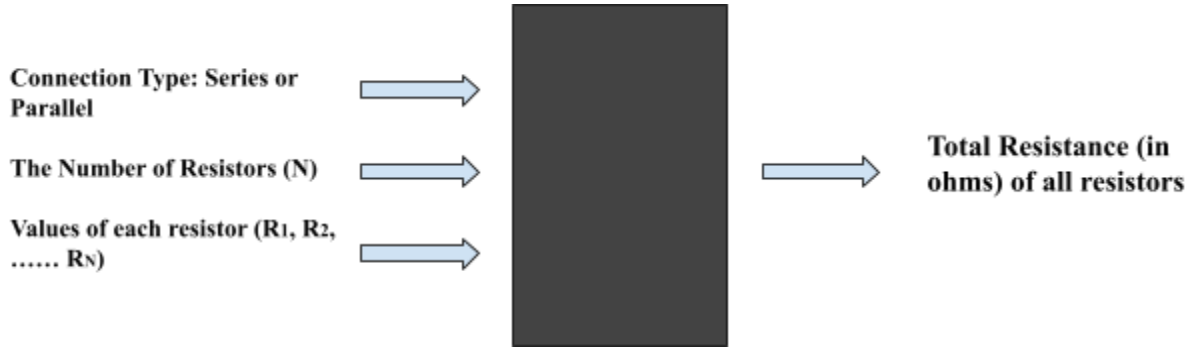


Figure 4: I/O Diagram

STEP 3: Test Cases and Algorithm Design

Two different types of connections, series and parallel, each have their own equation that should be used to calculate the total resistance in ohms. I will present each formula for finding the total resistance separately, accompanied by examples.

To find the total resistance for **Series Connection**:

1. Identify the resistors in the series connection and determine their resistance values. In the case below, there are three resistors each of which represent R1, R2, and R3 while they have resistance values of 4 (ohms), 7 (ohms), and 9 (ohms), respectively.
2. Add up all the resistance values: Rtot = R1 + R2 + R3
   In the case below, the total resistance will be 4 + 7 + 9 = 20 (ohms).

   *These input/output values serve as one test case.

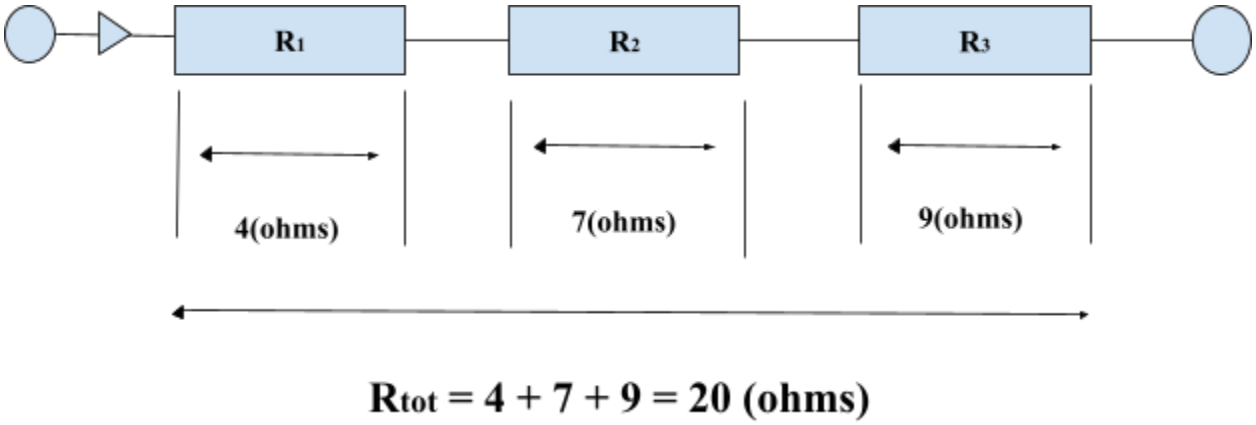


Figure 6: Finding Total Resistance for Series Connection

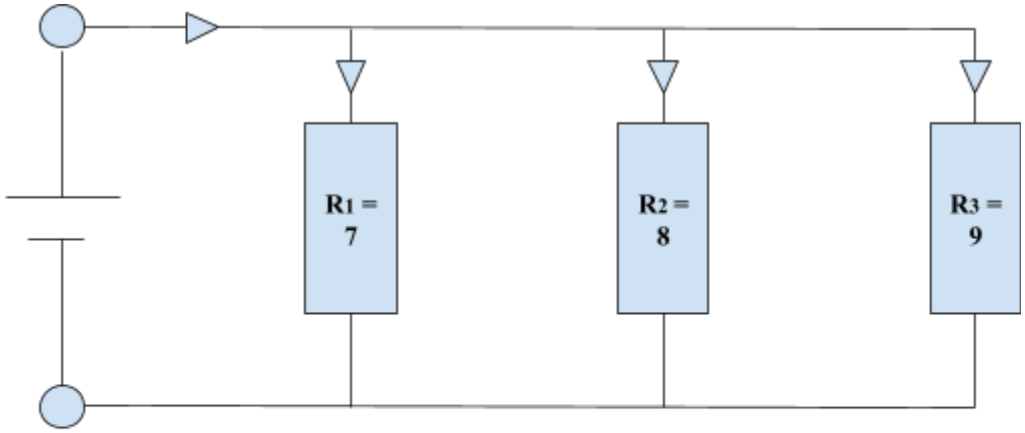
Thus, we can derive the formula for finding the total resistance of a series connection, which is illustrated below:

"$R_{tot} = R_1 + R_2 + R_3 + \cdots + R_N$ (where N is the number of resistors)"

To find the total resistance for **Parallel Connection**:

1. Identify the resistors in the parallel connection and determine their resistance values. In this case, there are three resistors each of which represent R1, R2, and R3 while they have resistance values of 7 (ohms), 8 (ohms), and 9 (ohms), respectively.
2. Take the reciprocal of each resistance value: $1/R_1$, $1/R_2$, $1/R_3$, ….$1/R_N$
   In this case, $1/R_1 = 1/7$, $1/R_2 = 1/8$, $1/R_3 = 1/9$
3. Add up all the reciprocals: $1/R_{tot} = 1/R_1+1/R_2+1/R_3+\cdots+1/R_N$
   In this case, $1/R_1+1/R_2+1/R_3 = 191/504$
4. Once you have the sum of the reciprocals, take the reciprocal of this sum to find the total resistance: $R_{tot} = 1 / (1 / R_1+1/R_2+1/R_3+\cdots+1/R_N)$
   In this case, $R_{tot} = 1/(191/504) =$ approx 2.64 (ohms)

   *These input/output values serve as one test case.



$1/R_{tot} = 1/7 + 1/8 + 1/9$
$1/R_{tot} = 191/504$
$R_{tot} = 504/191 =$ approx 2.64 (ohms)

Figure 7: Finding Total Resistance for Parallel Connection

Thus, we can derive the formula for finding the total resistance of a parallel connection, which is illustrated below:

"$1/R_{tot} = 1/R_1 + 1/R_2 + 1/R_3 + \cdot + 1/R_N$ (where N is the number of resistors)"

The following provides a set of test cases that can be used to test the algorithm and software. It is advisable to verify the program using a variety of numbers, excluding negative ones, ranging from small values to very large ones for both the number of resistors and resistance values, and also using negative values to check if the program will fail or not.

Series
Case 1) N = 3, R1 = 4, R2 = 7, R3 = 9 → The expected total resistance will be Rtot = 20
Case 2) N =10, R1 = 30, R2 = 657, R3 = 70, R4 = 64, R5 = 27, R6 = 100, R7 = 747, R8 = 62, R9 = 70, R10 = 673 → The expected total resistance will be Rtot = 2500
Case 3) N = 2, R1 = 57, R2 = -78 → The expected outcome will be invalid.
*Case 3 is to make sure that resistance value cannot be negative.

Parallel
Case 3) N = 3, R1 = 7, R2 = 8, R3 = 9 → The expected total resistance will be 1/Rtot = approx 2.64
Case 4) N = 12, R1 = 30, R2 = 87, R3 = 70, R4 = 94, R5 = 27, R6 = 1727, R7 = 743, R8 = 62, R9 = 70, R10 = 673, R11 = 737, R12 = 988 → The expected total resistance will be 1/Rtot = approx 2.64

Algorithm
The algorithm can be expressed as:
Declare mainChoice, resisNum = 0, resisValue = 0.0 as integers,

```
Repeat
        Print "Main menu"
        Print "1) Calculate total resistance value"
        Print "2) Help"
        Print "3) Exit"
        Print "Select your choice: "
        Read value into mainChoice

        If mainChoice is equal to 1,
                Print "Secondary Menu"
                Print "1) Series"
                Print "2) Parallel"
                Print "Choose the connection type"
                Declare secondaryChoice as integers and Initialize it to 0
                Read value into secondaryChoice

                If secondaryChoise is equal to 1 or 2,
                        Print "Please enter the total number of resistors: "
                        Read value into numResis
```

Declare totalResistance as integers and initialize it to 0
Declare totalInverseResistance as double and initialize it to 0.0

Repeat for each resistor in the range from 1 to numResis (initialize integer i as 1 and i will increment by 1 each time until it reaches the numResis)
Repeat
Print “Please enter the value of resistor”, i, and “(Omega)”
Read value into resisValue

If resisValue is smaller than 0,
Print “Your answer is invalid. Resistance value cannot be negative. Please try again.”
While logical expression, which is resistance is smaller than 0, is true

if secondaryChoice is equal to 1,
Add resisValue to totalResistance
Otherwise
Add 1 / resisValue to totalResistance

If sceondaryChoice is equal to 1,
Print “The total resistance value is: “, totalResistance, and “(Omega)”
Otherwise
Declare totalParallelResistance
Assign (1 / totalResistance) to totalParallelResistance
Print “The total resistance value is: “, totalParallelResistance, and “(Omega)”

Otherwise
Print “Your answer is invalid. Please try again.”

Otherwise if mainChoise is equal to 2,
Print “This program allows you to calculate the total resistance value of a set of resistors.”
Print “You can choose between calculating in series or parallel.”
Print “Please follow the instructions provided in each menu.”

Otherwise if mainChoise is equal to 3,
Print “Program is terminated. Bye!”

Otherwise
Print “Your answer is invalid. Please try again.”

While logical expression, which is mainChoice is not equal to 3, is true

Return to 0

STEP 4: Implementation

```
/*-------------------------------------------------------------------------------------
-------*/
/* Name: Shota Matsumoto, Student Number: sm11745*/
/* Date: Feburary 24, 2024*/
/* Program: CPE First Assignment.cpp*/
/* Description: This program computes the total resistance in series and parallel
connections*/
/*
-------------------------------------------------------------------------------------
----*/
#include <iostream>
using namespace std;
int main() {
    int mainChoice = 0;
    int numResis = 0;
    int resisValue = 0;
    //Insert do/while Repetition structure to evaluate the condition at the end
    do {
        //Print Main Menu and three choices and Read choice
        cout << "Main Menu" << endl;
        cout << "1) Calculate total resistance value" << endl;
        cout << "2) Help" << endl;
        cout << "3) Exit" << endl;
        cout << "Please enter your choice: ";
        cin >> mainChoice;
        //Make if statement. Here we deal with the case of choice 1. Print two
connection types and Read choice
        if (mainChoice == 1) {
            cout << "Secondary Menu" << endl;
            cout << "1) Series" << endl;
            cout << "2) Parallel" << endl;
            cout << "Please select connection type: ";
            int secondaryChoice = 0; //declare and initialize variable as integer and
equal to 0
            cin >> secondaryChoice;
            //Make if statment for the case of secnodary choice being either 1 or 2.
Print the total number of resistors and read it
            if (secondaryChoice == 1 || secondaryChoice == 2) {
                cout << "Please enter the total number of resistors: ";
                cin >> numResis;
                //Declare and initialize variables
                int totalResistance = 0;
                double totalInverseResistance = 0.0;
                //Insert for looping structure. Declare and Initialize i which is a
counter, Provide conditions to the counter, and post-increment 1 every round
                for (int i = 1; i <= numResis; ++i) {
                                    do {
                                        cout << "Please enter the value of resistor "
<< i << " (Omega): ";
```

```
                                    cin >> resisValue;
                                    if (resisValue < 0) {
                                        cout << "Your answer is invalid.
Resistance value cannot be negative. Please try again." << endl;
                                    }
                                } while (resisValue < 0);


                    //Make if statement in the case of secondaryChoice being 1 and
Provide equation for series connection.
                    if (secondaryChoice == 1) {
                        totalResistance += resisValue;
                    }
                    //Install else to provide equation for parallel connection.
                    else {
                        totalInverseResistance += 1.0 / resisValue;
                    }
                }
                //Make if statement for the secondaryChoice being 1 and Print the
total resistance value.
                if (secondaryChoice == 1) {
                    cout << "The total resistance value is: " << totalResistance << "
(Omega)" << endl;
                }
                //Make else statement, Declare variable, and Provide equation to get
the total resistance for parallel connection. Print the total resistance value.
                else {
                    double totalParallelResistance;
                    totalParallelResistance = 1.0 / totalInverseResistance;
                    cout << "The total resistance value is: " <<
totalParallelResistance << " (Omega)" << endl;
                }
            }

            //Make else statement for the case where the user inputs values other than
1 or 2. Print the input is valid.
            else {
                cout << "Your answer is invalid. Please try again." << endl;
            }
        }

        //Make else/if statement for the mainchoice being 2. Print the information
about the program and re-display main menu.
        else if (mainChoice == 2) {
            cout << "This program allows you to calculate the total resistance value
of a set of resistors." << endl;
            cout << "You can choose between calculating in series or parallel." <<
endl;
            cout << "Please follow the instructions provided in each menu." << endl;
        }

        //Make else/if statement for the choice being 3. This terminates the program.
        else if (mainChoice == 3) {
            cout << "Program is terminated. Bye!" << endl;
        }
```

```
        //Make else statement for values other than 1, 2, or 3. Allow users to input
the value again from main menu.
        else {
            cout << "Your answer is invalid. Please try again." << endl;
        }
    } while (mainChoice != 3); //Provide the condition for do/while Repepition
structure
    return 0;
}
```

STEP 5: Tests and Verification

Test Case 1:

The number of resistors is 3, and values of each resistor are R1 = 4, R2 = 7, and R3 = 9.

The output is 20 (Omega).

This is in agreement with the expected outcome, which is 20.

```
Secondary Menu
1) Series
2) Parallel
Please select connection type: 1
Please enter the total number of resistors: 3
Please enter the value of resistor 1 (Omega): 4
Please enter the value of resistor 2 (Omega): 7
Please enter the value of resistor 3 (Omega): 9
The total resistance value is: 20 (Omega)
```

Figure 8: The output for test case 1

Test Case 2:

The number of resistors is 10, and values of each resistor are R1 = 30, R2 = 657, R3 = 70, R4 = 64, R5 = 27, R6 = 100, R7 = 747, R8 = 62, R9 = 70, R10 = 673.

The output is 2500 (Omega).

This is in agreement with the expected outcome, which is 2500.

```
Please enter the value of resistor 3 (Omega): 70
Please enter the value of resistor 4 (Omega): 64
Please enter the value of resistor 5 (Omega): 27
Please enter the value of resistor 6 (Omega): 100
Please enter the value of resistor 7 (Omega): 747
Please enter the value of resistor 8 (Omega): 62
Please enter the value of resistor 9 (Omega): 70
Please enter the value of resistor 10 (Omega): 673
The total resistance value is: 2500 (Omega)
```

Figure 9: The output for test case 2

Test Case 3:

The number of resistors is 2, and values of each resistor are R1 = 57 and R2 = -78.

The output is “Your answer is invalid. Resistance value cannot be negative. Please try again.”

This is in agreement with the expected outcome, which is invalid.

```
Secondary Menu
1) Series
2) Parallel
Please select connection type: 1
Please enter the total number of resistors: 2
Please enter the value of resistor 1 (Omega): 57
Please enter the value of resistor 2 (Omega): -78
Your answer is invalid. Resistance value cannot be negative. Please try again.
Please enter the value of resistor 2 (Omega):
```

Test Case 4:
The number of resistors is 3, and values of each resistor are R1 = 7, R2 = 8, R3 = 9.
The output is 2.63874 (Omega).
This is in agreement with the expected outcome, which is approx 2.64.

```
Secondary Menu
1) Series
2) Parallel
Please select connection type: 2
Please enter the total number of resistors: 3
Please enter the value of resistor 1 (Omega): 7
Please enter the value of resistor 2 (Omega): 8
Please enter the value of resistor 3 (Omega): 9
The total resistance value is: 2.63874 (Omega)
```

Figure 10: The output for test case 3

Test Case 5:
The number of resistors is 12, and values of each resistor are R1 = 30, R2 = 87, R3 = 70, R4 = 94, R5 = 27, R6 = 1727, R7 = 743, R8 = 62, R9 = 70, R10 = 673, R11 = 737, R12 = 988.
The output is 2.63874 (Omega).
This is in agreement with the expected outcome, which is 2.64.

```
Secondary Menu
1) Series
2) Parallel
Please select connection type: 2
Please enter the total number of resistors: 3
Please enter the value of resistor 1 (Omega): 7
Please enter the value of resistor 2 (Omega): 8
Please enter the value of resistor 3 (Omega): 9
The total resistance value is: 2.63874 (Omega)
```

Figure 11: The output for test case 4

Conclusion

Based upon the outcomes of these 5 various test cases, we can conclude that the program functions successfully and solves for the total resistance in series and parallel connections.

User's Guide

- To execute the program, compile and run the code found in the file named CPE First Assignment.cpp

1. Main Menu:
   - You will be presented with the main menu upon running the program.
   - Three choices are available:
     1) Calculate total resistance value: Allows you to compute the total resistance in series or parallel connections.
     2) Help: Provides information about the program and its functionality.
     3) Exit: Terminates the program.

2. Calculate total resistance value:
   - After choosing option 1 from the main menu, you will be prompted to select the type of connection:
     1) Series: Computes the total resistance in a series connection.
     2) Parallel: Computes the total resistance in a parallel connection.
   - Enter your choice accordingly.

3. Resistor Input:
   - Upon selecting the connection type, you will be asked to enter the total number of resistors.
   - For each resistor, input the resistance value in ohms (Ω).

4. Result Display:
   - After entering the resistance values for all resistors, the program will calculate and display the total resistance value.
   - For series connections, the total resistance value is displayed directly.
   - For parallel connections, the program computes the total resistance using the reciprocal formula and displays the result.

5. Invalid Inputs:
   - If you input values other than 1, 2, or 3 in the main menu, the program will prompt you to try again.
   - Similarly, if you provide an invalid choice in the secondary menu, the program will request you to try again.

6. Help:
   - Option 2 in the main menu provides information about the program's purpose and instructions for usage.

7. Exiting the Program:
   - Selecting option 3 from the main menu terminates the program.