
# LABORATORY 5

March 2, 2021

Abstract

We learned to use multiple functions in order to perform calculations accurately and to format the codes. We additionally incorporated switch statements.

Jordan Xu

Ece 1310 C for Engineers

1. Write a program to find the equivalent resistance of an arbitrary series parallel resistive network. In this task:
    a. Data is entered as an integer pair: <Code-Value> <Resistor-Value>
    b. The resistors are scanned from right to left.

    Where
    - Code value 0 indicates starting resistor (right most resistor).
    - Code value 1 indicates the resistor is connected in series.
    - Code value 2 indicates the resistor is connected in parallel.
    - Code value -1 indicates the end of data entry.

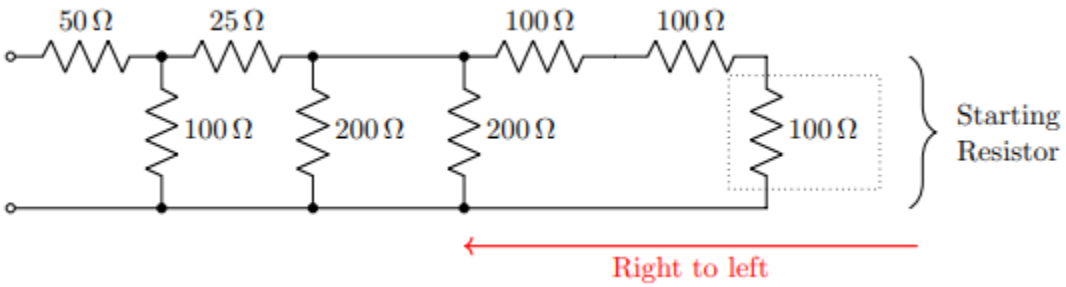



Figure 1: Resistive Network

The resistive network shown in Figure 1 will be presented to your program as shown below:

```
0 100
1 100
1 100
2 200
2 200
1 25
2 100
1 50
-1
```

As soon as the user enters -1, your program should print the value of the equivalent resistance for the circuit entered. It is easy to verify that the equivalent resistance for the circuit shown in Figure 1 is 100 Ω using Equations 1 and 2.

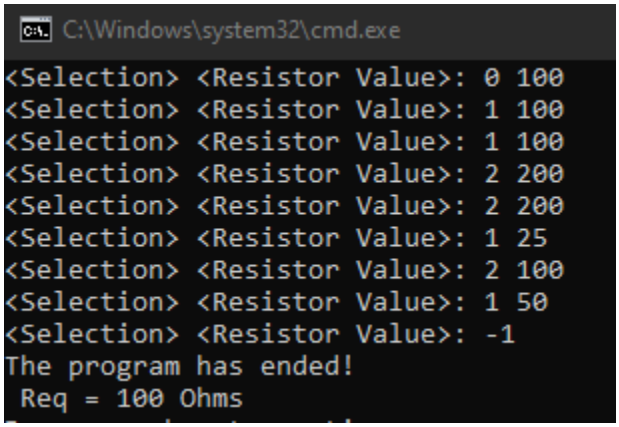
$$R_{eq} = R_1 + R_2 + R_3 + \cdots + R_n \qquad \text{(Series)} \qquad (1)$$

$$R_{eq} = (1/R_1 + 1/R_2 + 1/R_3 + \cdots + 1/R_n)^{-1} \qquad \text{(Parallel)} \qquad (2)$$

```
// ECE 1310-04
// Author: Jordan Xu
// Description: calculating resistances of a circuit
// Date: 5/2/2021
#include <iostream>
using namespace std;
void resitiveNetwork(int, int&, double&);
```

```
void invalidSection();
void flushBuffer();
static bool start = false; //understand purpose but not how it works

int main()

{
	int selection = 0, Rin; //Integer pair, initialize selection to zero to get into the
while loop.
	double Req = 0; //Initialize to zero in case the user wants to end the program
	while (selection != -1) // ends calculation if enter -1
	{
		cout << "<Selection> <Resistor Value>: ";
		cin >> selection;
		resitiveNetwork(selection, Rin, Req);
	}
	return 0;
}

void resitiveNetwork(int option, int& R, double& Rtot)
{
	switch (option) // determines if invalid, starting resistor, series, or parallel
	{
	case 0: // initial resistor
		if (start == false)
		{
			cin >> R;
			if (cin) // Checks for invalid values for R, e.g. $, %, etc.
			{
				Rtot = R;
				start = true;
			}
			else
				invalidSection();
		}
		else
			invalidSection();
		break;
	case 1: // series
		if (start)
		{
			cin >> R;
			if (cin) // Checks for invalid values
				Rtot += R;
			else
```

```
                                invalidSection();
                        }
                        else
                                invalidSection();
                        break;
                case 2: // parallel
                        if (start)
                        {
                                cin >> R;
                                if (cin) // Checks for invalid values
                                        Rtot = Rtot * R / (Rtot + R); // uses previous resistor
                                else
                                        invalidSection();
                        }
                        else
                                invalidSection();
                        break;
                case -1:
                        cout << "The program has ended!" << endl;
                        cout << " Req = " << Rtot << " Ohms" << endl;
                        break;
                default: invalidSection();
                }
                // In case of invalid resistor value, flush cin buffer
                flushBuffer(); // this is black magic
}

void invalidSection()
{
        cout << "Invalid selection" << endl;
}

void flushBuffer() // lost a brain cell here
{
        cin.clear(); // Clear cin flag
        cin.ignore(numeric_limits<streamsize>::max(), '\n'); // flush cin buffer
}
```


C:\Windows\system32\cmd.exe
<Selection> <Resistor Value>: 0 100
<Selection> <Resistor Value>: 1 100
<Selection> <Resistor Value>: 1 100
<Selection> <Resistor Value>: 2 200
<Selection> <Resistor Value>: 2 200
<Selection> <Resistor Value>: 1 25
<Selection> <Resistor Value>: 2 100
<Selection> <Resistor Value>: 1 50
<Selection> <Resistor Value>: -1
The program has ended!
Req = 100 Ohms