EECE.2160: ECE Application Programming
Spring 2018
Programming Assignment #2: Basic I/O and Operations
Due **Wednesday, 2/7/18**, 11:59:59 PM

1. Introduction

This assignment will give you some experience working with C input (using `scanf()`) and output (using `printf()`), as well as arithmetic operations with variables. You will do some very basic circuit analysis, reading the values of a voltage source and three resistors, and calculating the voltage across and current through each of the resistors using three different circuit configurations.

2. Deliverables

Submit your source file by uploading it directly to your Dropbox folder. Ensure your source file name is **_prog2_circuit.c_**. <u>Submit only the .c file.</u> Failure to meet this specification will reduce your grade, as described in the program grading guidelines.

3. Specifications

In this program, you will deal with the three resistive circuits in Figure 1: a series circuit, a parallel circuit, and a circuit using resistors both in series and in parallel.

Those of you who are familiar with basic DC circuit analysis should be able to easily derive the voltage across and current through each resistor. For more details and equations, please see Section 6: Circuit Analysis Supplement.
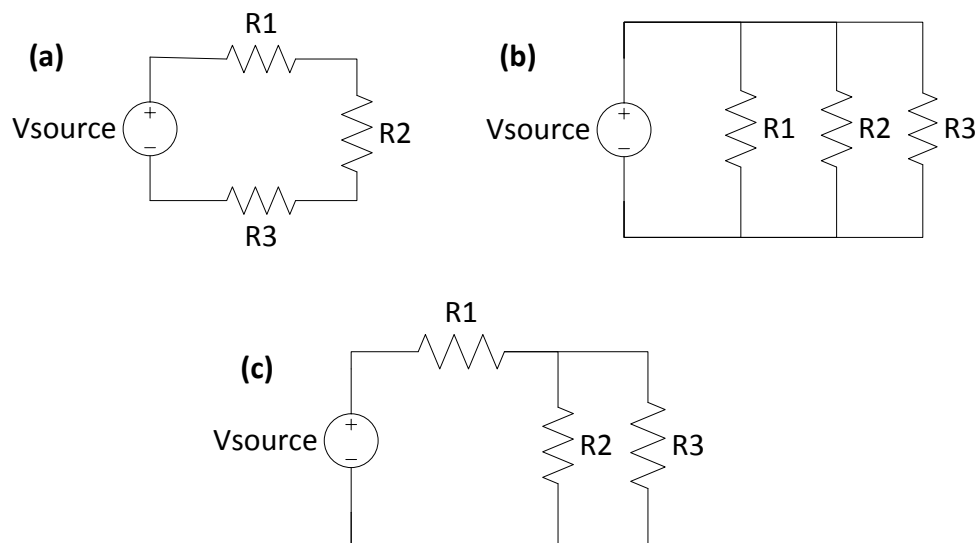

Figure 1: Basic three-resistor circuits, showing **(a)** all three resistors in series, **(b)** all three resistors in parallel, and **(c)** two of the resistors in parallel, with a third in series with each.

Input: Your program should prompt the user to enter the values of Vsource (in volts) and R1, R2, and R3 (in ohms). The source voltage should be entered on one line, while all three resistance values should be entered on a second line. (Note: remember, `scanf()` ignores whitespace when scanning numbers—you do not have to explicitly worry about varying numbers of spaces.)

The program could produce the first two lines below (user inputs are underlined):

```
Enter voltage source value (V):  10
Enter three resistance values (ohms):  5 10 10
```

All input values should be treated as double-precision floating point values.

Output: Once the voltage and resistance values have been read, you should print the following information for each circuit, as shown below:

- A blank line (to separate each circuit description)
- A brief circuit description (e.g., "SERIES CIRCUIT")
- The voltage across each resistor. Note that, in the parallel circuit, the voltage across each resistor is the same and should only be printed once.
- The current through each resistor. Note that, in the series circuit, the current through each resistor is the same and should only be printed once.

See Section 6 for more details on determining the appropriate voltage and current through each resistor if you are unfamiliar with the analysis used. The output lines that would follow the example shown above would be:

```
SERIES CIRCUIT
Current through circuit:  0.400000 A
Voltage across R1:  2.000000 V
Voltage across R2:  4.000000 V
Voltage across R3:  4.000000 V

PARALLEL CIRCUIT
Voltage across each resistor:  10.000000 V
Current through R1:  2.000000 A
Current through R2:  1.000000 A
Current through R3:  1.000000 A

R2 & R3 IN PARALLEL
Voltage across R1:   5.000000 V
Current through R1:   1.000000 A
Voltage across R2:   5.000000 V
Current through R2:  0.500000 A
Voltage across R3:   5.000000 V
Current through R3:  0.500000 A
```

See Section 5: Test Cases for more sample program runs.

4. Hints

Order of operations: C operators follow the same order of operations you likely learned for basic arithmetic operations—multiplication and division take precedence over addition and subtraction. Parentheses have higher precedence than any of these operators. So, for example, if $x = 5$, $y = 2$, and $z = 3$:

- `x + y * z   = 5 + 2 * 3   = 5 + 6 = 11`
- `(x + y) * z = (5 + 2) * 3 = 7 * 3 = 21`
- `x + y / 2   = 5 + 2 / 2   = 5 + 1 = 6`

Variables and expressions: Variables should be used to store values that are used more than once to avoid repeating calculations. For example, you could improve the following code by creating a variable to hold the value `a + 2`:

```
x = (a + 2) * 3;
y = 5 / (a + 2);
z = (a + 2) - (a + 2);
```

Values used only for output do not need variables, since `printf()` can print the value of any expression. Each of the following lines is therefore a valid use of this function. Assume you have variables `int n` and `double x`:

- `printf("n squared: %d, n cubed: %d\n", n * n, n * n * n);`
- `printf("17/x + 35n = %lf\n", (17 / x) + (35 * n));`
- `printf("Rectangle with length %d, width %lf has area %lf\n",`
    `n, x, n * x);`

Visual Studio scanf() error: Visual Studio users may encounter an error when using `scanf()` that says the function is unsafe. The message then suggests using `scanf_s()` instead, a Windows-specific secure version of `scanf()`.

I'd prefer you not use `scanf_s()`, as it's not a portable function (i.e., code that can be run on any system regardless of operating system), and it doesn't behave exactly the same way as `scanf()`. Instead, to resolve this error, include the following #define statement as shown—placing the define before the given include statement is strictly necessary:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

5. Test Cases

Your output should match these test cases exactly for the given input values. I will use these test cases to grade your program, but will also generate test cases that will not be publicly available. Note that these test cases may not cover all possible program outcomes. You should create your own tests to help debug your code and ensure proper operation for all possible inputs.

```
C:\Windows\system32\cmd.exe

Enter voltage source value (V):   20
Enter three resistance values (ohms):   10 3 8

SERIES CIRCUIT
Current through circuit:  0.952381 A
Voltage across R1:   9.523810 V
Voltage across R2:   2.857143 V
Voltage across R3:   7.619048 V

PARALLEL CIRCUIT
Voltage across each resistor:   20.000000 V
Current through R1:   2.000000 A
Current through R2:   6.666667 A
Current through R3:   2.500000 A

R2 & R3 IN PARALLEL
Voltage across R1:   16.417910 V
Current through R1:   1.641791 A
Voltage across R2:   3.582090 V
Current through R2:   1.194030 A
Voltage across R3:   3.582090 V
Current through R3:   0.447761 A

Press any key to continue . . .
```

```
C:\Windows\system32\cmd.exe

Enter voltage source value (V):   1.23
Enter three resistance values (ohms):   4.5 6.7 8.9

SERIES CIRCUIT
Current through circuit:  0.061194 A
Voltage across R1:   0.275373 V
Voltage across R2:   0.410000 V
Voltage across R3:   0.544627 V

PARALLEL CIRCUIT
Voltage across each resistor:   1.230000 V
Current through R1:   0.273333 A
Current through R2:   0.183582 A
Current through R3:   0.138202 A

R2 & R3 IN PARALLEL
Voltage across R1:   0.665070 V
Current through R1:   0.147793 A
Voltage across R2:   0.564930 V
Current through R2:   0.084318 A
Voltage across R3:   0.564930 V
Current through R3:   0.063475 A

Press any key to continue . . .
```

Remember, if you are using Visual Studio or Visual C++ Express, to get your program to terminate with a message saying, "Press any key to continue …", use the **Start Without Debugging** command (press Ctrl + F5) to run your code.

6. Circuit Analysis Supplement

The basic equations needed for this program are shown below in Figure 2. For more details on the analysis of series/parallel circuits, contact your instructor directly.
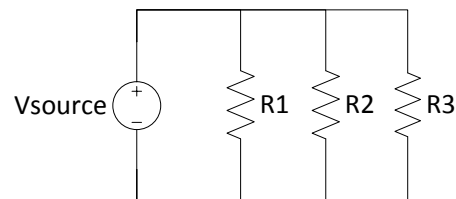
In all cases, given x = 1, 2, or 3:
· Vx = voltage drop across resistor Rx
  (for example, V1 = voltage across R1)
· Ix = current through resistor Rx
  (for example, I3 = current through R3)

SERIES CIRCUIT
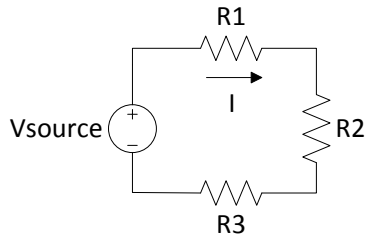
R_{total} = R1 + R2 + R3
I   = Vsource / R_{total}
         = I1 = I2 = I3 *(all currents equal)*
Vx   = I * Rx

PARALLEL CIRCUIT

Vsource = V1 = V2 = V3 *(voltage across all resistors equal)*
Ix = Vsource / Rx

SERIES/PARALLEL COMBINATION

To solve for all voltages/currents, convert the original circuit to the equivalent circuit on the right
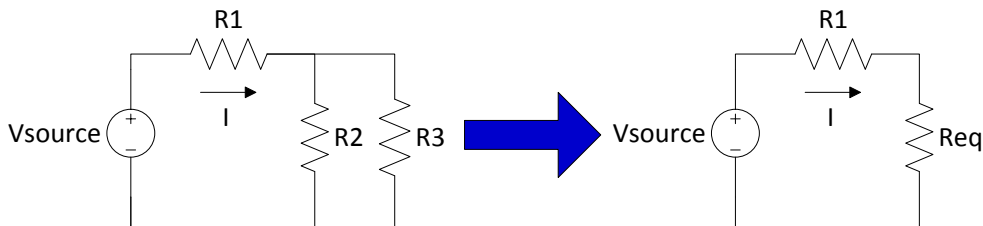
Req   = single resistance equivalent to parallel combination of R2, R3
      = (R2 * R3) / (R2 + R3)
I   = Vsource / (R1 + Req)
Veff = I * Req

V1   = I * R1
V2 = V3 = Veff   *(voltage across parallel resistors is the same)*
I1   = I
I2   = Veff / R2
I3   = Veff / R3

Figure 2: Circuit diagrams with equations for analyzing each circuit