A. Resistor combinations

Prepare a C program that does the following:

1. Declares three integer variables, resistor1, resistor2, and resistor3, and two double variables, series and parallel. You can use different names, if you prefer. You can use more variables in program, if you think you need them.
2. Uses scanf() to allow the user You) to enter values for the three resistors. Include proper prompts (using printf) so that the user knows what is to be entered. For the purposes of our program, reasonable resistor values are between 100 Ω and 10,000 Ω, but they must integers.
3. Calculates the series and parallel equivalent resistances of the three resistors.
4. Outputs the entered resistor values and corresponding values of the series and parallel combinations. Include enough text and formatting so that the user can easily understand what is being displayed. This includes listing the correct units.

Note: If you don't yet know your resistor combinations, resistors in series add directly $R_S = \Sigma R_n$, where R_n represents the various values of the individual resistors. For the parallel combination, the inverses of the resistors add to give the inverse of the equivalent resistance: $(R_P)^{-1} = \Sigma(R_n)^{-1}$.

B. Power and energy

Prepare a C program that does the following:

1. Declares two double variables: energy, power, and one integer variable: hours. You can use different names, if you prefer. You can use more variables in program, if you think you need them.
2. Uses scanf() so that the user can enter values for power and hours. Include proper prompts (using printf) so that the user know what is being entered. For the purposes of our program, reasonable powers are between 1 W and 10,000 W. and reasonable times are between 1 hour and 168 hours (= 1 week).
3. Calculates the total energy that was expended as the power was used during that time.
4. Outputs the entered power and time and the calculated energy. Include enough text and formatting so that the user can easily understand what is being displayed. This includes listing the correct units.

Note: Of course, power is energy/time. The units of energy are joules and the units of power are joules/sec. If you want to get fancy, try expressing the energy in terms of kW-hours, which are the units by the power company when they send you the electric bill.

C. Ohm's Law and resistor energy

Prepare a C program that does the following:

1. Declares three integer variables: voltage, resistance, and hours and three double variables: current power, and energy. You can use different names, if you prefer. You can use more variables in program, if you think you need them.
2. Uses scanf() so that the user (you) can enter values for voltage, resistance, and hours. Include proper prompts (using printf) so that the user knows what is being entered. For the purposes of our program, reasonable of voltage are between 1 and 120 V, reasonable values of resistance are from 10 Ω to 10,000 Ω, and reasonable values of time are between 1 hour and 168 hours.
3. Calculates the current that flows in the resistor if entered is applied across it (using Ohm's law), the power being dissipated in the resistor with that amount of current flowing, and total amount of energy converted to heat in the resistor during the time that was entered.
4. Outputs the entered voltage, resistance, and time along with the calculated current, power ,a and energy. Include enough text and formatting so that the user can easily understand what is being displayed. This includes listing the correct units.

D. Finally, there will a short quiz which will be very similar to the in-class quizzes, except that this time you will enter the program and run it on your computer. (In fact, it may very well be one of the in-class quizzes.) The quiz this week will cover variable statements and calculations. You will demo the quiz to your instructor and print out a hard copy of the program.

"Reporting"

The three programs (A, B, and C above) can be written prior to lab. You can demo the programs on your own laptop brought to lab or using VirtualBox in the lab room — it's your choice. Your instructor will first ask you to demonstrate one of the programs. A correctly functioning program will earn 15 points. The instructor will then ask you to make one or two modifications to your program. Then you will demo the modified program. If you successfully modify the program, then you will earn an extra 10 points for a total of 25 for the program. You should then print out and submit a written copy of your original program. (You can print out your original version prior to lab, if you are 100% sure that it works properly.)

The process is then repeated for the second and third programs, which are also worth 25 points each.

The quiz is worth 25 points, so that the entire lab is worth 100 points.