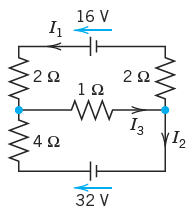
# For this homework assignment, you are encouraged to work in groups of two or three. No groups larger than three please.

# Analysis of a resistor network: A program called HW6\_1.py has been created to analyze the resistor networks pictured below. Study both the text file ResistorNetwork.txt (for the left circuit) and modify it to include a 5Ω resistor in parallel with the 32V source and this as ResistorNetwork\_2.txt. Modify the necessary classes in Resistor.py, Loop.py, VoltageSource.py and ResistorNetwork.py to analyze both circuits. For the right circuit below, we use *inheritance* to create a class called ResistorNetwork\_2 that inherits all the attributes and methods from ResistorNetwork. Demonstrate *polymorphism* by rewriting AnalyzeCircuit and GetKirchoffVals functions to appropriately handle the second circuit.

**Notes:**

1. This object oriented program has been divided into several files in the HW6\_1 directory and some of the code contains #JES Missing Code, that you must fix.
2. Kirchoff current law states that the net current flowing into a node must be zero.
3. Kirchoff voltage law states that the net voltage drop around a closed loop in a circuit must be zero.
4. In analyzing a circuit, we imagine traversing a loop in one direction (e.g., counter clockwise). The voltage may either increase or decrease across a circuit element in the traversal direction depending on the direction of current flow in that element. For example, if I1 is in the direction as shown in the left circuit and we are traversing in a counterclockwise direction, we would expect the voltage to decrease from node b to c.



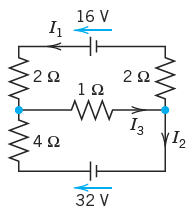
a

b

c

d

e



a

b

c

d

e

*I5*



*I4*

f

# 2. A pipe network problem: Write an object oriented program called HW6\_2 that uses FSOLVE() to find the volumetric flow rate (m3/s) of water in each segment of the pipe flow network given below.

Use the following properties:

water density = 1000 kg/m3

water viscosity = 0.00089 N⋅s/m2

pipe roughness = 0.00025 m.

Your program should print the flow in each segment of pipe, nicely formatted, similar to:

The flow in segment a-b is -0.0052 m^3/s

The flow in segment a-c is 0.0272 m^3/s

The flow in segment d-g is -0.0142 m^3/s

etc.

Notes:

1. Pressure decreases in direction of flow (e.g., )
2. The head loss around a pipe loop is zero. (e.g.,)
3. Mass is conserved at each node.
4. Pressure loss in a pipe is calculated with the Darcy-Weisbach equation:
5. Darcy friction factor is found as in your part b from Homework 5 (i.e., if laminar: use 64/Re, if turbulent: use the Colebrook equation, if transitional: use *normalvariant*)
6. A program stem for this problem has been uploaded to Canvas that is object oriented.
7. Analyzing a pipe network is equivalent to analyzing a resistor network:
   1. No net flow of fluid into a node.
   2. No net pressure loss around a closed loop.
   3. Traversal direction around a loop and the flow direction of fluid needs to be considered to determine if pressure rises or falls in the direction of traversal.
8. As in problem 1, the work of problem 2 has been split across several files and #JES Missing Code helps you find where the code is broken.



# 3. A Rankine Power Cycle - OOP

Note: steam-stem.py and rankine-stem.py have been written and uploaded to github for your use.

You will be writing three python files – steam.py, rankine.py and test\_rankine.py used to analyze two different Rankine power cycles:

1. p\_high=8000kPa, p\_low=8kPa, x1=1 (i.e., saturated vapor entering turbine)
2. same as *i*) except that T1=1.7⋅ Tsat (i.e., superheated steam into the turbine).

In both cases, the turbine and pump operate isentropically.

The file test\_rankine.py should import from rankine.py and instantiate two different rankine objects with the properties as specified above. Calculate the cycle efficiencies and output a report for each cycle.

**For this problem, you should use scipy.interpolate.griddata to find thermodynamic properties. You may check your answers with PyXSteam.**

Chart, histogram

Description automatically generated