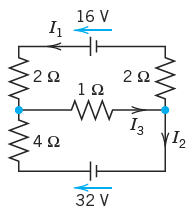
# For this homework assignment, you are encouraged to work in groups of two or three. No groups larger than three please.

# 1. Analysis of a resistor network: A program called HW6\_1\_OOP.py has been created to analyze the resistor network pictured below. Study both the text file ResistorNetwork.txt and modify it to include a 5Ω resistor in parallel with the 32V source. Save this as ResistorNetwork\_2.txt. Next, modify HW6\_1\_OOP.py to analyze this modified circuit in a program saved as: HW6\_1\_2\_OOP.py. In this second part of the problem, import all the classes from HW6\_1\_OOP.py and use *inheritance* to create a class called ResistorNetwork2 that inherits all the instance variables and methods from ResistorNetwork. Demonstrate *polymorphism* by rewriting AnalyzeCircuit and GetKirchoffVals functions to appropriately handle the second circuit.



a

b

c

d

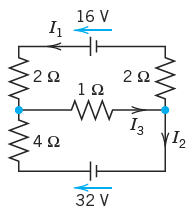
e

*I5*



*I4*

f



a

b

c

d

e

# 2. A pipe network problem: Write an object oriented program called HW6\_2\_OOP 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.



# 3. A Rankine Power Cycle - OOP

Note: steam-stem.py and rankine-stem.py have been written and uploaded to Canvas 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.

Chart, histogram

Description automatically generated