6.01 Midterm 2 Spring 2011

Name:	Solutions	Section:

These solutions do not apply for the conflict exam.

Enter all answers in the boxes provided.
Clearly written work will be graded for partial credit.

During the exam you may:

- read any paper that you want to
- use a calculator

You may not

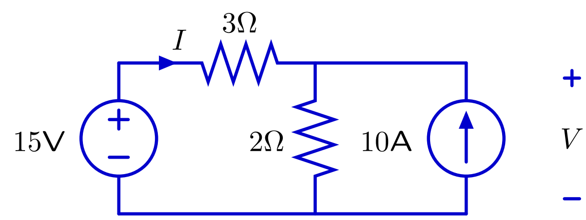
- use a computer, phone or music player

For staff use:

1.	/12
2.	/12
3.	/12
4.	/12
5.	/18
6.	/12
7.	/12
8.	/10
total:	/100

1 Find the Voltage and Current (12 points).

Determine V and I in the following circuit.



V = 18V

I = −1A

Use superposition.

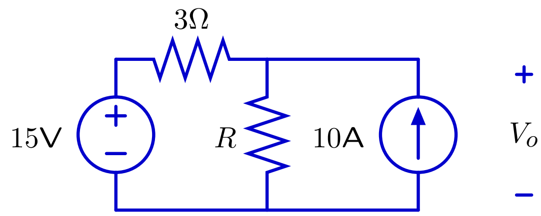
If the current source is set to zero (replaced with an open circuit), then V would be $2\Omega/(2\Omega + 3\Omega) \times 10V= 6$V and I would be $15\text{V}/(3\Omega + 2\Omega) = 3$A.

If the voltage source is set to zero (replaced by a short circuit), then V would be $10\text{A}\times(2\Omega||3\Omega) = 10 \times (2 \times 3)/(2+3) = 12$V and I would be $-2\Omega/(2\Omega + 3\Omega) \times 10A= -4$A.

The sums are $\text{V} = 6 + 12 = 18$V and $\text{I} = 3 - 4 = -1$A.

2 Find the Resistance (12 points).

Find the value of R so that $V_o = 30V$.



Enter your answer below, or enter **none** if no such value of R can be found.

R = 6Ω

By Ohm's law, the current through R is

$$I_R = \frac{V_o}{R} .$$

Applying KCL to the node where R joins the current source and the 3Ω resistor,

$$I_R = 10A - \frac{V_o - 15V}{3\Omega} .$$

Equating these, and setting $V_o = 30V$ yields $R = 6\Omega$.

3 LTI SM (12 points).

Write a difference equation for each of these machines if it describes an LTI system or give a very brief reason why it does not. The input to the machine at step n is $x[n]$ and the output of the machine at step n is $y[n]$.

```
class MM1(sm.SM):
   startState = [0, 0]
   def getNextValues(self, state, inp):
       return ([state[1], inp], 2*state[0])
```

$$y[n] = 2x[n-2]$$

```
class MM2(sm.SM):
   startState = [0]
   def getNextValues(self, state, inp):
       return (state + [inp], sum(state))
```

$$y[n] = y[n-1] + x[n-1]$$

```
class MM3(sm.SM):
   startState = 0
   def getNextValues(self, state, inp):
       return (max(state, inp), max(state, inp))
```

Max is not a linear operator.

```
class MM4(sm.SM):
   startState = 0
   def getNextValues(self, state, inp):
       return (state + 1, state)
```

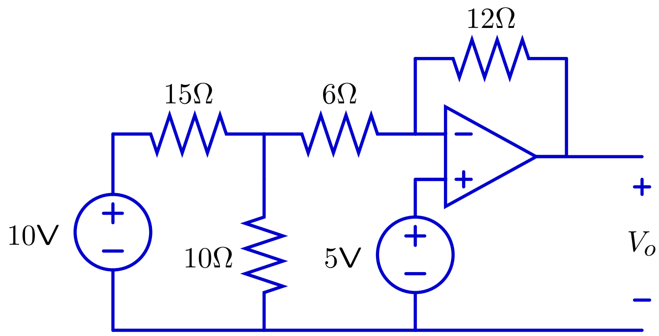
It is tempting to write:

$$y[n] = y[n-1] + 1$$

but the constant term is not legal in a difference equation.

4 Op-Amp Circuit (12 points).

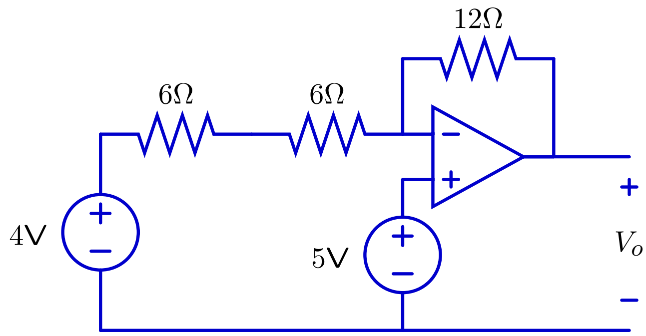
Determine V_o in the following circuit. Assume that the op-amp is ideal.



V_o =

6V

Express the left voltage source and two left-most resistors as a Thevenin equivalent, with Thevenin voltage $\frac{10}{10+15}$10V= 4V and Thevenin resistance $10\Omega || 15\Omega = 10 \times 15/(10+15) = 6\Omega$.



Since $V_- = V_+$, $V_- = 5$V. So there must be 1/12A flowing left through the two 6Ω resistors. There must be a corresponding 1/12A flowing to the left through the 12Ω resistor, since no current enters the V_- input of the op-amp. V_o is then the sum of $V_- = 5$V and the 1V across the 12Ω resistor.

5 Run Length (18 points).

One simple approach to sequence compression is called *run-length encoding* (RLE). A *run* is a subsequence of repeated entries. The idea is to represent the original sequence by a list of pairs of the form:

```
(runLength, entry)
```

For example, we could represent this list of digits:

```
[3, 3, 3, 3, 5, 5, 9, 9, 9, 3, 3]
```

by this:

```
[(4, 3), (2, 5), (3, 9), (2, 3)]
```

This representation is useful when there are likely to be long subsequences of repeated entries in the sequence.

In this problem, you will define a class to represent and manipulate RLE sequences.

```
class RLE:
    def __init__(self, seq):
        self.rleSeq = self.encode(seq)
    def encode(self, seq):
        # code 1
    def decode(self):
        # code 2
    def add(self, other):
        # code 3
```

5.1 Encoding

Write the definition of the encode method, which takes a list of digits and returns an RLE-encoded list.

```
def encode(self, seq):

        rle = []
        prev = None
        count = 0
        for x in seq:
            if x == prev: count = count+1
            else:
                if prev: rle.append((count, prev))
                prev = x
                count = 1
        if prev: rle.append((count, prev))
        return rle
```

5.2 Decoding

Write the definition of the decode method, which returns a list of digits corresponding to the RLE-encoded list for the class instance.

```
def decode(self):

        seq = []
        for (count, entry) in self.rleSeq:
            for i in xrange(count):
                seq.append(entry)
        return seq
```

5.3 Addition

Let's define addition on our sequences as component-wise addition. Assume that both sequences are the same number of characters when decoded.

```
>>> RLE([2,3,4,4,4]).add(RLE([2,3,3,3,4]))
```

should produce a new instance of the RLE class whose content is:

```
[(1, 4), (1, 6), (2, 7), (1, 8)]
```

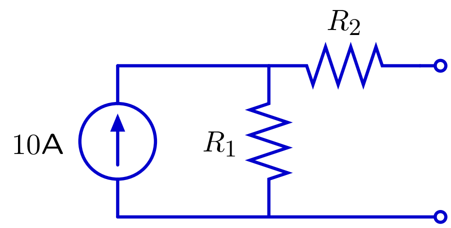
Don't try to be efficient in your solution. It's fine to decode the sequences to add them.

```
def add(self, other):

        seq1 = self.decode()
        seq2 = other.decode()
        return RLE([x + y for (x, y) in zip(seq1, seq2)])
```

6 Make it Equivalent (12 points).

Determine values of R_1 and R_2 in the following circuit



so that

- the Thevenin equivalent voltage $V_T = 1V$, and
- the Thevenin equivalent resistance $R_T = 1\Omega$.

$R_1 =$ $\frac{1}{10}\Omega$

$R_2 =$ $\frac{9}{10}\Omega$

The Thevenin voltage is the open-circuit voltage:

$$V_T = R_1 \times 10A = 1V.$$

Thus $R_1 = \frac{1}{10}\Omega$.

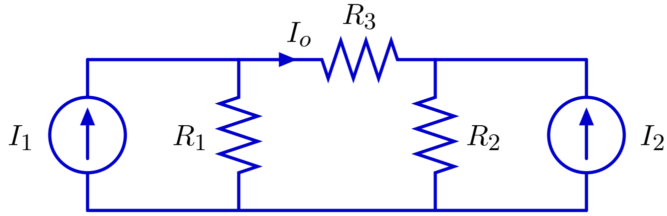
The Thevenin resistance is V_T over the short-circuit current:

$$R_T = \frac{1V}{\frac{R_1}{R_1+R_2}10A} = 1\Omega.$$

Solving, we get $R_2 = \frac{9}{10}\Omega$.

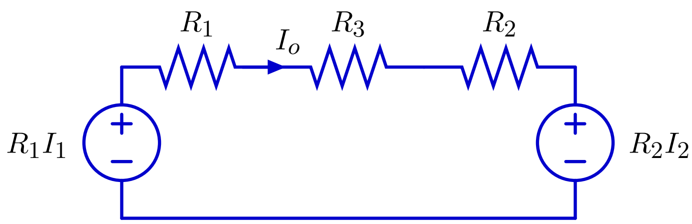
7 Current from Current Sources (12 points)

Determine an expression for I_o in the following circuit.



$$I_o = \frac{R_1 I_1 - R_2 I_2}{R_1 + R_2 + R_3}$$

Replace the part of the circuit that contains I_1 and R_1 with its Thevenin equivalent; then do the same with the part that contains I_2 and R_2:



Now the resistors are in series, as are the voltage sources:

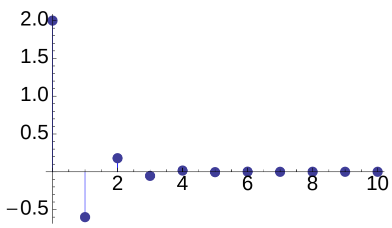
$$I_o = \frac{R_1 I_1 - R_2 I_2}{R_1 + R_2 + R_3}$$

8 Poles (10 points)

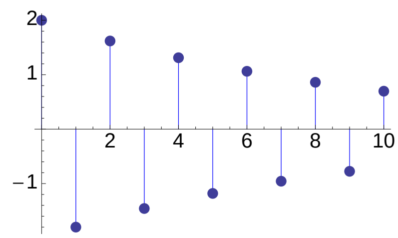
Each signal below has the form

$$s[n] = (a + bj)^n + (a - bj)^n$$

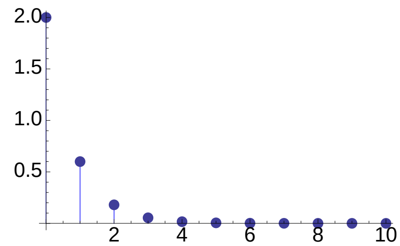
where a and b can have values $0, 0.3, 0.5, 0.9, 1.1, -0.3, -0.5, -0.9, -1.1$. The periodic signals have a period of either 2, 4, or 8. For each one, specify a and b.



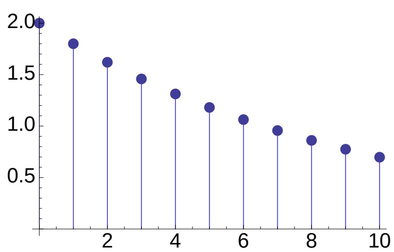
$a : -0.3 \;\; b : 0$



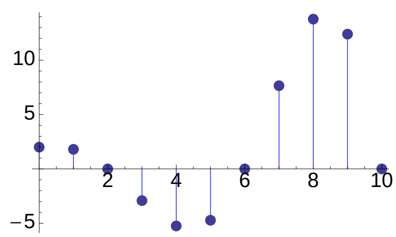
$a : -0.9 \;\; b : 0$



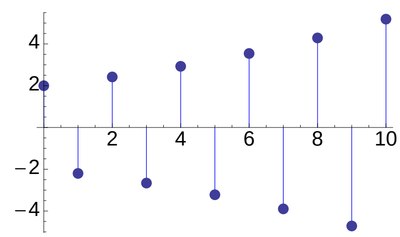
$a : 0.3 \;\; b : 0$



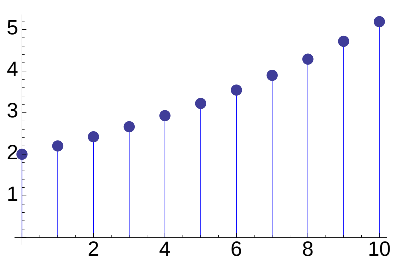
$a : 0.9 \;\; b : 0$



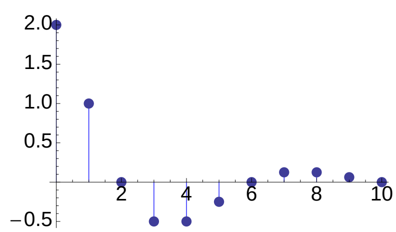
$a : 0.9 \;\; b : \pm 0.9;$



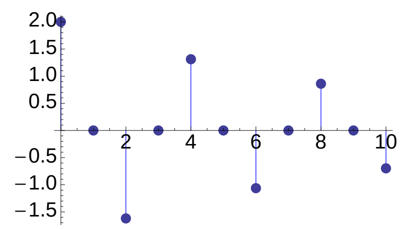
$a : -1.1 \;\; b : 0$



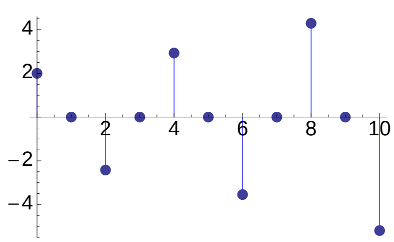
$a : 1.1 \;\; b : 0$



$a : 0.5 \;\; b : \pm 0.5$



$a : 0.0 \;\; b : \pm 0.9$



$a : 0.0 \;\; b : \pm 1.1$

Worksheet (intentionally blank)