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:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>/12</td>
</tr>
<tr>
<td>2.</td>
<td>/12</td>
</tr>
<tr>
<td>3.</td>
<td>/12</td>
</tr>
<tr>
<td>4.</td>
<td>/12</td>
</tr>
<tr>
<td>5.</td>
<td>/18</td>
</tr>
<tr>
<td>6.</td>
<td>/12</td>
</tr>
<tr>
<td>7.</td>
<td>/12</td>
</tr>
<tr>
<td>8.</td>
<td>/10</td>
</tr>
<tr>
<td>total:</td>
<td>/100</td>
</tr>
</tbody>
</table>
1 Find the Voltage and Current (12 points).
Determine \( V \) and \( I \) in the following circuit.

\[
\text{Diagram: Circuit with a 15V battery, 3\( \Omega \) resistor, 2\( \Omega \) resistor, and 10A current source.}
\]

\( V = \) 

\( I = \)
2 Find the Resistance (12 points).

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

![Circuit Diagram]

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

$$R =$$
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 \( y[n] \).

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

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

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

```python
class MM4(sm.SM):
    startState = 0
    def getNextValues(self, state, inp):
        return (state + 1, state)
```
4 Op-Amp Circuit (12 points).

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

$V_o =$
5 Run Length (18 points).

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

\[(\text{runLength, entry})\]

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.

```python
def encode(self, seq):
```
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):

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.

```python
>>> 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:

```python
[(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.

```python
def add(self, other):
```

6 Make it Equivalent (12 points).

Determine values of $R_1$ and $R_2$ in the following circuit

\begin{align*}
\text{so that} \\
\bullet \text{ the Thevenin equivalent voltage } V_T &= 1 \text{V, and} \\
\bullet \text{ the Thevenin equivalent resistance } R_T &= 1 \Omega.
\end{align*}

\begin{align*}
R_1 &= \\
R_2 &= 
\end{align*}
7 Current from Current Sources (12 points)

Determine an expression for $I_o$ in the following circuit.

$I_0 =$
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 \( b \).