The nature of the problem is suited to a computational solution

The four steps can be described as follows :

1. Fill the larger container.

2. Pour the larger one into the smaller one, until the smaller container is full or the larger one is empty.

3. Empty the small one into a sink.

4. Pour the difference from the larger into the smaller.

Repeat steps 1-4. The larger container will be empty and the smaller one will contain 4 litres.

Writing your Thinking

Take 5 minutes to think about how you tackle this problem.

- **Did you use pen/paper/whiteboard to help figure possible solutions?**
- Did you draw diagrams to represent the problem?
- **Steps 2 and 4 involve pouring form large to small. But they are slightly different situations. How are they different?**
- Did you use Trial and Error to investigate possible solutions?
- **Did you verify your solution?**

Using Think-Pair-Share-Square (TPSS), go through how you and your partner were thinking about how to solve the problem.

Abstraction

We need to set the capacities of the 2 containers (5 and 7 litres in this case).

We also need to keep track of the volumes in each container before and after each step of the algorithm.

Some of the key components of the problem are :

- A container object must have a maximum volume and current volume field. (A dictionary is ideally suited to this).
- Filling is setting the current volume to the max of the container.
- Emptying is setting the current volume to zero.
- Pouring from the large to small container could involve a large container that is full or partially filled and a small container that is empty or partially filled.

Pseudo-Code

```
#Initialise the conditions
Create a small container; Set maxV = 5 and currentV = 0;
Create a large container; Set maxV = 7 and currentV = 0;
For 2 iterations {
        #Print out the state of each container at each step
        #Step1 Set large_currentV = 7;
        #Step2 Add the large_currentV on to small_currentV;
                Set the large_currentV = zero;
                #In doing this transfer we may have overflowed the small
                If small_currentV > small_maxV {
                    Put the surplus back into large_currentV;
                    Set small_currentV to its maxV;
                }
        #Step3 Set small_currentV = 0;
        #Step4 Execute the step 2 mini-algorithm again;
}
Output the final result;
```