1. **Overrunning a loop:**

   Write an m-file **Lsum1.m** that will input a number L and add up enough terms of the series

   $$\frac{1}{3}+\frac{1}{6}+\frac{1}{9}+\frac{1}{12}+\frac{1}{15}+\dots$$

   until the sum reaches L or greater. Try $L = 1,\ 2,\ 3.$

   *The m-file should display the number of terms (**k**) needed as well as the actual sum, call it **mysum**.*

   Ans. mysum = 1.0066; k = 11 mysum = 2.0015; k = 227 mysum = 3.0001; k = 4550

   Theoretically this sum adds up to infinity, but its gets there very very slowly. With a moderate L, say $L \geq 7$ *your computer might take a while to produce an answer see item (4) below.*

   *To keep track of how the addition is progressing (and to check that the computer has not gone to sleep) you could print out the sum, say every 10,000 terms. If you called your sum **mysum**, then include the following in your while loop:*

```
if rem(k, 10000) == 0
    disp(['the sum after ', num2str(k) ,' terms is ', num2str(mysum)])
end
```

   *Of course this will make the loop take longer, so it's a trade-off between speed and knowing how far you have got. Now run **Lsum1.m** with* $L = 5$.

   Ans. mysum = 5; k = 1,835,421

2. **Stopping overruns:**

   Modify your m-file above (**Lsum1.m** and call it **Lsum2.m**) so that it does not overrun − the sum must not exceed L. *[Hint: Use an **if - break** construct inside the while loop.]*

   Make it display the counter (**k**), the last term (**term**) and the sum (**mysum**) at every step. *[Hint: Inside your while loop, add the following two lines at appropriate positions]*

```
disp('        k       term       mysum')
disp([k term mysum])
```

   Thus here we want the final printed sum to be $\leq L$. Try it with $L = 1/2,\ 3/2,\ 9/5.$

   Ans. k = 2, mysum = 0.5 k = 50, mysum = 1.4997 k = 123, mysum = 1.7978

3. **Overrunning to stop an overrun?**

   Sometimes it is better to let a loop overrun and then 'step back' after the loop has ended, i.e. below the loop.

   - subtract **term** from **mysum**,
   - subtract 1 from the counter **k**,
   - re-calculate the actual last term from the corrected value of **k**.

   Modify **Lsum2.m** and call it **Lsum3.m** so that it applies these ideas.

   We still want to make it display the counter (**k**), the last term (**term**) and the sum (**mysum**) at every step. Try it on $L = 1/2,\ 3/2,\ 9/5.$

4. **Test your looping skills:**

   (a) Plot the graph of V against t, where

   $$V(t) = 1 + \sum_{n=1}^{M} (-1)^{n-1} \frac{4}{(2n-1)\pi} \cos\left(\frac{(2n-1)\pi}{2} t\right).$$

   Here $M = 100$ and t is a vector with 500 elements starting at $t = 0$ and ending at $t = 25$.

   (b) Plot the graph of P against t, where

   $$P(t) = \pi + \sum_{k=1}^{N} \frac{(-1)^{k-1}}{\exp(k/5)} \sin(k\,t).$$

   Here **N** is the smallest integer such that the addition of any extra term to the sum will not change the MATLAB value of the sum (i.e. the smallest number of terms after which the sum becomes constant) and t is a vector with 500 elements starting at $t = 0$ and ending at $t = 25$.