CS1112 Section Exercise 3

1 Multiples of k

The following program reads an integer k and outputs all positive multiples of k up to 1000. For example, if $k = 150$, then 150, 300, 450, 600, 750, and 900 would be displayed. Fill in the blank.

```
k = input('Please enter a positive integer smaller than 1000:  ');

for j = ______________________
   fprintf('%d ', j);
end
fprintf('\n');
```

2 Approximate π

[Modified from *Insight* Exercise P2.1.5] For large n,

$$T_n = 1 + \frac{1}{2^2} + \cdots + \frac{1}{n^2} = \sum_{k=1}^{n} \frac{1}{k^2} \approx \frac{\pi^2}{6}$$

$$R_n = 1 - \frac{1}{3} + \cdots - \frac{(-1)^{n+1}}{2n-1} = \sum_{k=1}^{n} \frac{(-1)^{k+1}}{2k-1} \approx \frac{\pi}{4}$$

giving two different ways to estimate π:

$$\tau_n = \sqrt{6T_n}$$
$$\rho_n = 4R_n$$

Write a script that displays the value of $|\pi - \rho_n|$ and $|\pi - \tau_n|$ for $n = 100 : 100 : 1000$ in one table.

3 The one-million-digit $n!$

If the value of `x` is a positive integer, then the value of `floor(log10(x))+1` is the number of digits in its base-10 expansion. For example, if $x = 123$, then $\log_{10}(x) = 2.0899$. The floor of that is 2 and the recipe says 123 requires 2+1 digits. (Side question. Why don't we use `ceil(log10(x))`?)

Noting that

$$\log_{10}(n!) = \log_{10}(2) + \log_{10}(3) + \cdots + \log_{10}(n)$$

write a script that prints out the smallest n so that $n!$ has at least one million digits. This is a `while`-loop problem. You'll need a running sum to add up all the logs. The `while`-loop condition will involve checking the status of that sum. Where does the `fprintf` statement go?