Statistics 406 Problem Set 1

Due in lab, Tuesday September 19

1. Write a short R program to calculate the number of digits in a positive integer `x`. There are several ways to solve this problem, but for one approach you may find the identity

$$\log_a(x) = \log_b(x)/\log_b(a)$$

to be useful. Also remember that the `log` function in R is the natural (base e) logarithm. You may also need to use the `ceiling` function, which rounds a real number toward ∞ (for reference, the `floor` function rounds toward $-\infty$). There is also a solution to this problem not involving logarithms. You only need to give one correct solution.

Solution: The basic idea is that x has k digits if and only if

$$10^{k-1} \leq x < 10^k.$$

For example, x has two digits if and only if

$$10^1 = 10 \leq x < 100 = 10^2.$$

Therefore,

$$k - 1 \leq \log_{10} x < k,$$

so `1+floor(log(x)/log(10))` is one possible solution. An equivalent solution using `ceiling` is:

```
ndigit <- ceiling(log(x+1)/log(10))
```

Here is a solution not using logarithms:

```
k <- 0
while (x >= 1)
{
  x <- x/10
  k <- k+1
}
```

2. Implement the 'sieve of Erasthones' to determine whether an integer x is prime. This involves checking each integer y between 2 and $\sqrt{x}$ to determine whether y evenly divides x. Your program should construct an integer variable f such that at the end of execution, the value of f is 1 if x is prime and 0 if it is composite.

   Solution:

```
f <- 1
for (y in 2:sqrt(x))
{
  if (y == x) { break }
  if (x %% y == 0)
  {
    f <- 0
    break
  }
}
```

3. Building on your solution to problem 2, write a program to count the number of prime integers between 2 and 10000.

   Solution: I get 1229 primes using the following program:

```
nprime <- 0

for (x in 2:10000)
{
  f <- 1
  for (y in 2:sqrt(x))
  {
    if (y == x) { break }
    if (x %% y == 0)
    {
      f <- 0
      break
    }
  }

  if (f == 1) { nprime <- nprime+1 }
}
```