

```
1 #lang racket
2 ; Returns the result of raising base to the (non-negative)
2 power exp.
3 (define (pow base exp)
4   (if (< exp 1)
5       1
6       (* base (pow base (- exp 1)))))
7
8
9 ; Returns n!.
10 (define (factorial n)
11
12
13
14
15
16
17
18 ; Returns the nth number in the Fibonacci series.
19 (define (fib n)
20
21
22
23
24
25
26
27 ; Returns #t if n is prime and #f otherwise.
28 ; The first function shown is a helper function.
29 (define (prime?-help n i)
30
31
32
33
34
35
36
37
38
39 (define (prime? n) (prime?-help n 2))
```