

```
1 #lang racket
2
3 (define (GCD x y)
4   (if (= x 0)
5       y
6       (if (= y 0)
7           x
8           (GCD y (modulo x y)))))
9
10 (GCD 41 4)
11
12 (GCD 5740 70)
13
14 (define (GCD-2 x y)
15   (cond ((= x 0) y)
16         ((= y 0) x)
17         (else (GCD-2 y (modulo x y)))))
18
19 (GCD-2 5740 70)
20
21 (define (greet str)
22   (printf (string-append "hi " str)))
23
24 (define (greet-let str)
25   (let ((greeting (string-append "hi " str)))
26     (printf greeting))
27   )
28
29 (greet-let "Carolyn")
30
31 (define (GCD-3 n1 n2)
32   (cond ((= n1 0) n2)
33         ((= n2 0) n1)
34         (else (let ((remainder (modulo n1 n2)))
35                 (GCD-3 n2 remainder)))))
36
37 (GCD-3 48 6)
```