

```
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 ;; solution using cond
15
16 (define (GCD-2 x y)
17   (cond ((= x 0) y)
18         ((= y 0) x)
19         (else (GCD-2 y (modulo x y)))))
20
21 (GCD-2 5740 70)
22
23
24 ;; solution using let
25
26 (define (GCD-3 n1 n2)
27   (cond ((= n1 0) n2)
28         ((= n2 0) n1)
29         (else (let ((remainder (modulo n1 n2)))
30                 (GCD-3 n2 remainder)))))
31
32 (GCD-3 48 6)
```