```
(define (gcd a b)
  (if (= b 0)
      a
      (gcd b (remainder a b))))
(gcd 206 40)
```

During normal order, `if` is evaluated the same as in applicative order, meaning the predicate is evaluated first, and the result determines whether to evaluate the consequent or alternative. Normal-order evaluation only evaluates expressions when they are needed, and instead first substitutes operand expressions as the parameters until only primivate operators remain.
The following uses the substitution method to illustrate the process generated in evaluating (`gcd 206 40`) using normal-order evaluation. `*(remainder a b)*` indicates a `remainder` operation that is actually performed to reach the next step.

```
(gcd 206 40)
(if (= 40 0)
    206
    (gcd 40 (remainder 206 40)))
(gcd 40 (remainder 206 40)))
(if (= *(remainder 206 40)* 0)
    40
    (gcd (remainder 206 40) (remainder 40 (remainder 206 40))))
(if (= 6 0)
    40
    (gcd (remainder 206 40) (remainder 40 (remainder 206 40))))
(gcd (remainder 206 40) (remainder 40 (remainder 206 40))))
(if (= (remainder 40 *(remainder 206 40)*) 0)
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))))
(if (= *(remainder 40 6)* 0)
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))))
(if (= 4 0)
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))))
(gcd (remainder 40 (remainder 206 40))
     (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))))
(if (= (remainder *(remainder 206 40)* (remainder 40 *(remainder 206 40)*)) 0)
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))
(if (= (remainder 6 *(remainder 40 6)*) 0)
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))
(if (= *(remainder 6 4)* 0)
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))
(if (= 2 0)
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))
(gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
     (remainder (remainder 40 (remainder 206 40))
                (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))))
(if (= (remainder (remainder 40 *(remainder 206 40)*)
                  (remainder *(remainder 206 40)*
                             (remainder 40 *(remainder 206 40)*)))
       0)
    (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
    (gcd (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))
         (remainder (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
                    (remainder (remainder 40 (remainder 206 40))
                               (remainder (remainder 206 40)
                                          (remainder 40 (remainder 206 40)))))))
(if (= (remainder *(remainder 40 6)*
                  (remainder 6 *(remainder 40 6)*))
       0)
    (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
    (gcd (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))
         (remainder (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
                    (remainder (remainder 40 (remainder 206 40))
                               (remainder (remainder 206 40)
                                          (remainder 40 (remainder 206 40)))))))
(if (= (remainder 4 *(remainder 6 4)*) 0)
    (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
    (gcd (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))
         (remainder (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
                    (remainder (remainder 40 (remainder 206 40))
                               (remainder (remainder 206 40)
                                          (remainder 40 (remainder 206 40)))))))
(if (= *(remainder 4 2)* 0)
    (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
    (gcd (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))
         (remainder (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
                    (remainder (remainder 40 (remainder 206 40))
                               (remainder (remainder 206 40)
                                          (remainder 40 (remainder 206 40)))))))
(if (= 2 0)
    (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
    (gcd (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40) (remainder 40 (remainder 206 40))))
         (remainder (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
                    (remainder (remainder 40 (remainder 206 40))
                               (remainder (remainder 206 40)
                                          (remainder 40 (remainder 206 40)))))))
```

```
(remainder *(remainder 206 40)* (remainder 40 *(remainder 206 40)*))
(remainder 6 *(remainder 40 6)*)
*(remainder 6 4)*
```

2

During normal-order evaluation, 18 `remainder` operations are actually performed.

The following will produce the same, except for applicative-order evaluation, where the subexpressions are evaluated before application of the procedure which is the value of the operator.

```
(gcd 206 40)
(if (= 40 0)
    206
    (gcd 40 (remainder 206 40)))
(gcd 40 *(remainder 206 40)*)
(gcd 40 6)
(if (= 6 0)
    40
    (gcd 6 (remainder 40 6)))
(gcd 6 *(remainder 40 6)*)
(gcd 6 4)
(if (= 4 0)
    6
    (gcd 4 (remainder 6 4)))
(gcd 4 *(remainder 6 4)*)
(gcd 4 2)
(if (= 2 0)
    4
    (gcd 2 (remainder 4 2)))
(gcd 2 *(remainder 4 2)*)
(gcd 2 0)
(if (= 0 0)
    2
    (gcd 0 (remainder 0 2)))
```

2

During applicative-order evaluation, only 4 `remainder` operations are actually performed.