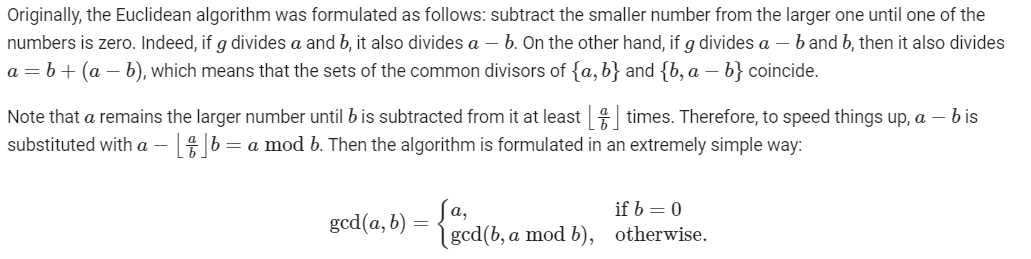
**Experiment-30**

**Aim**-Write a LISP Program using recursion to perform GCD of two numbers entered by user.

**Description**—Euclidean algorithm for computing the greatest common divisor



**Program**-

;;; Common Lisp implementation of GCD

;;; -------------------------------- The gcd function in LISP.

(defun gcd (a b)

(if (= b 0)

a

(gcd b (mod a b))))

;;; --------------------------------------------- Calling gcd

(defun gcdprint(a b)

(let (

(f (list "The gcd of " a " and " b " is " (gcd a b)))

)

(mapcar 'princ f)

(terpri)

)

)

;;; --------------------------------------------- Top-level

(defun run()

(princ "Greatest Common Divisor")

(terpri)

(loop

(princ "Enter two integers (0 0 to end): " )

(setq n1 (read)) ; A global. This is non-functional usage.

(setq n2 (read)) ; There is no need to use assignment in Lisp.

(cond ((= n1 0) (return))

(T (gcdprint n1 n2))

)

)

)

(run )

**Input**-

25

61

10

20

100

100

0

**Output**-

