```
┌──────────────────────────── MODULE Euclid ────────────────────────────┐
EXTENDS Integers, GCD
CONSTANTS M, N
 ****************************************************************************
--algorithm Euclid{
    variables x ∈ 1 . . N, y      ∈ 1 . . N, x0 = x, y0 = y ;
    { while ( x ≠ y ) { if ( x < y ) { y := y − x }
                               else { x := x − y }
                             } ;
      print ⟨x, y⟩ ;
      assert (x = y) ∧ (x = GCD(x0, y0)) ;
     }
 }
 ****************************************************************************
 BEGIN TRANSLATION (chksum(pcal) = "48d63aa1" ∧ chksum(tla) = "2425b90e")
VARIABLES x, y, x0, y0, pc

vars ≜ ⟨x, y, x0, y0, pc⟩

Init ≜  Global variables
         ∧ x ∈ 1 . . N
         ∧ y ∈ 1 . . N
         ∧ x0 = x
         ∧ y0 = y
         ∧ pc = "Lbl_1"

Lbl_1 ≜ ∧ pc = "Lbl_1"
          ∧ IF x ≠ y
                THEN ∧ IF x < y
                           THEN ∧ y′ = y − x
                                ∧ x′ = x
                           ELSE ∧ x′ = x − y
                                ∧ y′ = y
                     ∧ pc′ = "Lbl_1"
                ELSE ∧ PrintT(⟨x, y⟩)
                     ∧ Assert((x = y) ∧ (x = GCD(x0, y0)),
                              "Failure of assertion at line 11, column 7.")
                     ∧ pc′ = "Done"
                     ∧ UNCHANGED ⟨x, y⟩
          ∧ UNCHANGED ⟨x0, y0⟩

 Allow infinite stuttering to prevent deadlock on termination.
 Terminating ≜ pc = "Done" ∧ UNCHANGED vars

Next ≜ Lbl_1
          ∨ Terminating
```

$Spec \triangleq Init \land \Box[Next]_{vars}$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION

$Safety \triangleq (pc = \text{“Done”}) \Rightarrow (x = y) \land (x = GCD(x0, y0))$

\\ $*$ Modification History

\\ $*$ Last modified *Fri Sep* 24 19:56:42 *CST* 2021 by *wrz*

\\ $*$ Created *Fri Sep* 24 16:52:34 *CST* 2021 by *wrz*