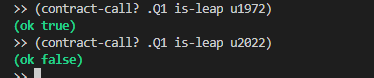
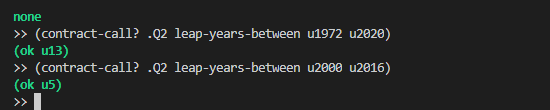
# **Q1**

;; Public functions  
(define-read-only (**is**-leap (year uint))  
 (**if** (**is**-eq (mod year u400) u0)  
 (ok true)  
 (**if** (**is**-eq (mod year u100) u0)  
 (ok false)  
 (**if** (**is**-eq (mod year u4) u0)  
 (ok true)  
 (ok false)  
 )  
 )  
 )  
)



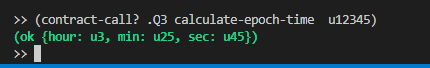
# **Q2**

(define-read-only (check-year (year uint))   
 (**if** (> year u0)  
 (let  
 (  
 (prev\_year (- year u1))  
 )  
 (- (/ prev\_year u4) (+ (/ prev\_year u100) (/ prev\_year u400)))  
 )  
 (- (/ year u4) (+ (/ year u100) (/ year u400)))  
 )  
)  
  
  
(define-read-only (leap-years-between (start uint) (end uint))  
 (**if** (< start end)  
 (ok (- (check-year end) (check-year (+ start u1))))  
 (ok u0)  
 )  
)



# **Q3**

((define-read-only (calculate-hours (time uint))   
 (/ time u3600)  
)  
  
(define-read-only (calculate-minutes (time uint))   
 (/ (mod (mod time u86400) u3600) u60)  
)  
(define-read-only (calculate-seconds (time uint))   
 (let   
 (  
 (last\_digits (mod time u100))  
 )  
 (**if** (< last\_digits u10)   
 (+ u00 last\_digits)  
 (+ last\_digits u0)  
 )  
 )  
)  
  
(define-read-only (calculate-epoch-time (time uint))   
 (let   
 (  
 (hours (calculate-hours time))  
 (minutes (calculate-minutes time))  
 (seconds (calculate-seconds time))  
 (result {hour: hours, min: minutes, sec: seconds})  
 )  
 (ok result)  
 )  
)



# **Q4**

(define-read-only (get-missing (numbers (list 10 uint)))  
 (let   
 (  
 (sum (fold + numbers u0)) ;; Adding all the integers **from** the list using fold  
 (missing-number (- u55 sum)) ;; Subtracting the sum **from** the sum of first 10 decimals  
 )   
 (ok missing-number)  
 )  
)

