1. **Write a read-only function that returns true if the given year is a leap year else return false**

**Prototype:**

// returns bool

(define-read-only (is-leap (year uint))

// your logic here

)

**Test Cases:**

**Input:** u1972 **Output:** True

**Input:** u2022 **Output:** False

**Note:** You can make helper functions but they must act read-only as well.

**Reference:**

[**https://www.programiz.com/cpp-programming/examples/leap-year**](https://www.programiz.com/cpp-programming/examples/leap-year)

[**https://docs.stacks.co/write-smart-contracts/language-functions#if**](https://docs.stacks.co/write-smart-contracts/language-functions#if)

1. **Write a read-only function that returns the number of leap years between two years.**

**Prototype:**

// returns uint

(define-read-only (no-leap (year1 uint) (year2 uint))

// your logic here

)

**Test Cases:**

**Input:** u1972, u2020 **Output:** u13

**Input:** u2000, u2016 **Output:** u5

**Note:** Your function must return the number of leap years inclusive of the two years passed as arguments

**Reference:**

[**https://stackoverflow.com/questions/4587513/how-to-calculate-number-of-leap-years-between-two-years-in-c-sharp**](https://stackoverflow.com/questions/4587513/how-to-calculate-number-of-leap-years-between-two-years-in-c-sharp)

[**https://miniwebtool.com/leap-years-list**](https://miniwebtool.com/leap-years-list)

1. **Write a read-only function that returns the hour, minute and second represented by the given Unix epoch timestamp in GMT.**

**Prototype:**

// returns tuple

(define-read-only (get-date (timestamp uint))

// your logic here

)

**Test Cases:**

**Input:** u12345 **Output:** {hour: u3, min: u25, sec: u45}

**Note:** You don’t need to calculate the year, month, day.

**Reference:**

[**https://www.jotform.com/help/443-mastering-date-and-time-calculation**](https://www.jotform.com/help/443-mastering-date-and-time-calculation/)

[**https://docs.stacks.co/write-smart-contracts/language-functions#mod**](https://docs.stacks.co/write-smart-contracts/language-functions#mod)

[**https://www.epochconverter.com**](https://www.epochconverter.com)

1. **Write a read-only function that returns the missing number from a list of first 10 natural numbers starting from 1.**

**Prototype:**

// returns uint

(define-read-only (get-missing (numbers (list 10 uint)))

// your logic here

)

**Test Cases:**

**Input:** (list u1 u2 u3 u4 u6 u7 u8 u9 u10) **Output:** u5

**Input:** (list u1 u3 u4 u5 u6 u7 u8 u9 u10) **Output:** u2

**Note:** There will only be one number missing and all other numbers will be present.

**Reference:**

<https://docs.stacks.co/write-smart-contracts/language-functions#fold>

**Usage:** (contract-call? .<contract-name> get-missing (list u1 …… u10))