1. **Write a read-only function that returns the non-repeating number from a list of numbers**

**Prototype:**

// returns uint

(define-read-only (get-non-repeating (numbers (list 10 uint)))

// your logic here

)

**Test Cases:**

**Input:** (list u1 u2 u1 u4 u6 u4 u6 u9 u9) **Output:** u2

**Input:** (list u1 u3 u4 u7 u3 u5 u4 u7 u5) **Output:** u1

**Note:** There is only one non-repeating number in the list. All other numbers repeat **exactly once at any position**.

**Hint:** Check out the XOR operator and its usage.

**Reference:**

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

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

1. **Write a read-only function that returns a string indicating whether a given number is even or odd**

**Prototype:**

// returns “even” or “odd”

(define-read-only (get-type (num uint))

// your logic here

)

**Test Cases:**

**Input:** u6 **Output:** “even”

**Input:** u5 **Output:** “odd”

**Note:** You are **not allowed** to you if-else, asserts or any other branch/control-flow statements. Your function must be branchless.

1. **Create a custom trait NFT that can only be minted once by each user/principal. Choose appropriate asset identifier for your NFT.**

**Note:** Your trait must include the following essential functions like transfer, burn, mint, get-owner.

1. **In a separate contract, create a public function “give-commission” that transfers STX to a certain principal. Call this contract function inside the transfer function of your NFT in Q#3**

**Note:** Choose the commission principal and transfer amount of your choice.

**Reference:**

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

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