```
use aiken/collection/list use cardano/assets.{PolicyId, quantity_of} use cardano/transaction.{Input, Output} use upgradable_multisig/types.{MultisigDatum, PubKeyHash}

pub const multisig_token_name = “multisig”

pub fn get_input_by_token( inputs: List, policy_id: PolicyId, token_name: ByteArray, ) -> Option { list.find( inputs, fn(input) { quantity_of(input.output.value, policy_id, token_name) > 0 }, ) }

pub fn validate_multisig_datum(datum: MultisigDatum) -> Bool { let signers_nr = list.length(datum.signers) (signers_nr >= datum.threshold)? && (datum.threshold > 0)? }

fn has_enough_signers( all_signers: List, threshold: Int, actual_signers: List, ) -> Bool { when all_signers is { [] -> False [head, ..tail] -> if list.has(actual_signers, head) { if threshold > 1 { has_enough_signers(tail, threshold - 1, actual_signers) } else { True } } else { has_enough_signers(tail, threshold, actual_signers) } } }

pub fn signed_within_threshold( datum: MultisigDatum, extra_signatories: List, ) -> Bool { has_enough_signers(datum.signers, datum.threshold, extra_signatories) }
```