// This assignment was not too terrible for me. I expected it to be when I read

// the description, but it turned out to be about a 2.5/5 difficulty, and I learned

// about how to access functions within a function and how those variables can connect.

const validator = (function () {

    // ...

    let isValid = true;

    return {

        isNumeric: function (text) {

            if (typeof text !== "number") {

                isValid = false;

            }

        },

        isInteger: function (text) {

            /\* ... \*/

            if (typeof text === "number" && text !== Math.floor(text)) {

                return;

            } else {

                isValid = false;

            }

        },

        isNegativeInteger: function (text) {

            /\* ... \*/

            if (

                typeof text === "number" &&

                text === Math.floor(text) &&

                text < 0

            ) {

                return;

            } else {

                isValid = false;

            }

        },

        isPositiveInteger: function (text) {

            /\* ... \*/

            if (

                typeof text === "number" &&

                text === Math.floor(text) &&

                text > 0

            ) {

                return;

            } else {

                isValid = false;

            }

        },

        isNonNegativeInteger: function (text) {

            /\* ... \*/

            if (

                typeof text === "number" &&

                text === Math.floor(text) &&

                text >= 0

            ) {

                return;

            } else {

                isValid = false;

            }

        },

        isInRange: function (text, m, n) {

            /\* ... \*/

            if (typeof text !== "number") {

                isValid = false;

                return;

            }

            if (m && n && m < n && text >= m && text <= n) {

                return;

            }

            if (m && n === "undefined" && text >= m) {

                return;

            }

            if (m === "undefined" && n && text <= n) {

                return;

            } else {

                isValid = false;

            }

        },

        isValidEmail: function (text) {

            /\* ... \*/

            let emailFormat =

                /(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])/;

            if (text.match(emailFormat)) {

                return;

            } else {

                isValid = false;

            }

        },

        isNonEmpty: function (text) {

            /\* ... \*/

            if (typeof text == "string" && !text) {

                isValid = false;

            }

        },

        matchesRegex: function (text, regex) {

            /\* ... \*/

            let regexFormat = regex;

            if (text.match(regexFormat)) {

                return;

            } else {

                isValid = false;

            }

        },

        lengthIsInRange: function (text, m, n) {

            /\* ... \*/

            if (typeof text !== "string") {

                isValid = false;

                return;

            }

            if (m && n && m < n && text.length >= m && text.length <= n) {

                return;

            }

            if (m && n === "undefined" && text.length >= m) {

                return;

            }

            if (m === "undefined" && n && text.length <= n) {

                return;

            } else {

                isValid = false;

            }

        },

        isValid: function () {

            /\* ... \*/

            return isValid;

        },

        reset: function () {

            /\* ... \*/

            isValid = true;

        },

        returnmessage: function () {

            if (validator.isValid()) {

                console.log("All is well");

            } else {

                console.log("Something failed validation");

            }

        },

    };

})();

validator.reset(); // Write some code to test your solution

validator.isNumeric(5); //this should work

validator.returnmessage();

validator.reset();

validator.isNumeric("5"); //this should NOT work

validator.returnmessage();

validator.reset();

validator.isPositiveInteger(5); //this should work

validator.returnmessage();

validator.reset();

validator.isNegativeInteger(-18); //this should work

validator.returnmessage();

validator.reset();

validator.isNonNegativeInteger(0); //this should work

validator.returnmessage();

validator.reset();

validator.isNonNegativeInteger(1); //this should work

validator.returnmessage();

validator.reset();

validator.isPositiveInteger(0); //this should NOT work

validator.returnmessage();

validator.reset();

validator.isInRange(5, 4, 6); //this should work

validator.returnmessage();

validator.reset();

validator.isInRange(4, 5, 6); //this should NOT work

validator.returnmessage();

validator.reset();

validator.isValidEmail("noeliakroot!!@gmail.com"); //this should work

validator.returnmessage();

validator.reset();

validator.isNonEmpty(); //this should NOT work

validator.returnmessage();

validator.reset();

validator.lengthIsInRange("hello", 2, 5); //this should work

validator.returnmessage();

validator.reset();

validator.matchesRegex("5.5", /^\d+$/); //this should NOT work

validator.returnmessage();

validator.reset();

validator.matchesRegex("5", /^\d+$/); //this should work

validator.returnmessage();

validator.reset();