Question1 : Write API:

Using Python Flask or ExpressJS, Write a REST API that reads the body and  returns JSON.

# API Method POST

# URL : /find\_symbols\_in\_names

# Input JSON Body of the API:

{

    "chemicals": ['Amazon', 'Microsoft', 'Google'],

    "symbols": ['I', 'Am', 'cro', 'Na', 'le', 'abc']

}

# Output: display the chemical names with their symbol surrounded by square brackets:

{

    "result": "[Am]azon, Mi[cro]soft, Goog[le]"

}

**Solution:**

**Index.js**

const express = require('express');

const app = express();

const bodyParser = require('body-parser');

const cors = require("cors");

const { json } = require('body-parser');

// Port

let port = process.env.PORT || 5000;

// Take the permission from express for json data (postman)

app.use(express.json());

// Use bodyParser

app.use(bodyParser.urlencoded({ extended: true }));

// to avoid cors header error

app.use(cors());

app.get('/find\_symbols\_in\_names', (req, res) => {

    // let obj = (req.body); // frontend data

    let obj = {

        chemicals: ['Amazon', 'Microsoft', 'Google'],

        symbols: ['I', 'Am', 'cro', 'Na', 'le', 'abc']

    };

    let array = [];

    obj.chemicals.map((chemical) => {

        obj.symbols.map((symbols) => {

            if (chemical.includes(symbols)) {

                array.push(chemical);

            }

        })

    })

    console.log(array);

    res.send(array);

})

// Listen

app.listen(port, () => {

    console.log(`Server started at port 5000`);

});

Question 2:

Given two arrays, write a function to compute their intersection.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2]

Output: [2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

Output: [9,4]

Note:

Each element in the result must be unique.

The result can be in any order.

**Solution:**

public int[] intersect(int[] nums1, int[] nums2) {

List<Integer> list = new ArrayList<Integer>();

for (Integer n : nums2) {

list.add(n);

}

int[] output = new int[list.size()];

int i = 0;

for (Integer s : nums1) {

if (list.contains(s)) {

output[i++] = s;

list.remove(list.indexOf(s));

}

}

// Initially we have taken result array size as list.size(). We are returning copy of array because the

// result array size can be smaller than list.size().

return Arrays.copyOf(output, i);

}

Question 3:

Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

Example 1:

Input: "()"

Output: true

Example 2:

Input: "()[]{}"

Output: true

Example 3:

Input: "(]"

Output: false

**Solution:**

public boolean isValid(String s) {

if(s.length() % 2 != 0) return false;

Stack<Character> stack = new Stack();

for(char c : s.toCharArray()){

if(c == '(' || c == '{' || c == '['){

stack.push(c);

}else if(c == ')' && !stack.isEmpty() && stack.peek() == '('){

stack.pop();

}else if(c == '}' && !stack.isEmpty() && stack.peek() == '{'){

stack.pop();

}else if(c == '[' && !stack.isEmpty() && stack.peek() == '['){

stack.pop();

}

}

return stack.isEmpty();

}

Question 4:

Given a non-empty array of integers, every element appears twice except for one. Find that single one.

Note:

Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

Example 1:

Input: [2,2,1]

Output: 1

Example 2:

Input: [4,1,2,1,2]

Output: 4

**Solution:**

public int singleNumber(int[] nums) {

int result = 0;

for(int i = 0; i<nums.length; i++){

result ^= nums[i];

}

return result;

}