**Data Structure and Algorithm Practicals**

4. Check for balanced parentheses by using Stacks

function isMatchingBrackets (str)

{

let stack = [];

let map = {

'(': ')',

'[': ']',

'{': '}'

};

for (let i = 0; i < str.length; i++) {

// If character is an opening brace add it to a stack

if (str[i] === '(' || str[i] === '{' || str[i] === '[' ) {

stack.push(str[i]);

}

// If that character is a closing brace, pop from the stack, which will also reduce the length of the stack each time a closing bracket is encountered.

else {

let last = stack.pop();

//If the popped element from the stack, which is the last opening brace doesn’t match the corresponding closing brace in the map, then return false

if (str[i] !== map[last]) {return false};

}

}

// By the completion of the for loop after checking all the brackets of the str, at the end, if the stack is not empty then fail

if (stack.length !== 0) {return false};

return true;

}

if ( isMatchingBrackets ("{]}"))

console.log("Correct");

else

console.log("INCorrect");