Graphical user interface, text

Description automatically generated

// you can write to stdout for debugging purposes, e.g.

// console.log('this is a debug message');

function solution(N) {

    // Implement your solution here

    for(var i = 0; i<1000000; i++){

        var str\_i=''+i

        var chararr\_i = str\_i.split('')

        var int\_arr = chararr\_i.map(function(x){

            return parseInt(x, 10);

        })

        if (SumAll(int\_arr) === N){

            return i

        }

    }

}

function SumAll(A){

    var result = 0

    for(var i = 0; i < A.length; i++){

        result += A[i]

    }

    return result

}

# Note: Incorrect solution

Text

Description automatically generated

Text

Description automatically generated

// you can write to stdout for debugging purposes, e.g.

// console.log('this is a debug message');

function solution(S) {

    S = S.split('');

    var sA  = true;

    var action = 0;

    var A=[]

    var B=[]

    var B\_l

    for(var i=0; i< S.length; i++){

        if(S[i]==='A'){

            A.push(i)

        } else {

            B.push(i)

        }

    }

    var check\_A = S[S.length/2-1] === 'A'

    var check\_B = S[S.length/2] === 'B'

    B\_l=B.length

    //This while block is just a cope

    while(A[A.length-1]>B[0]){

        if(B[0]<A[0]){

            B.shift()

            action++

            continue

        }/\* else if(A[A.length-1]>B[B.length-1]){

            A.pop()

        }\*/

        else{

            A.pop()

        }

        action++

    }

    //nothing about this conditional makes any sense.

    if(B.length === A.length && check\_A && check\_B){

        return B\_l

    }

    return action

}

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

// you can write to stdout for debugging purposes, e.g.

// console.log('this is a debug message');

function solution(A) {

    // Implement your solution here

    A = A.sort(function(a,b){return b-a})

    var reserveTrees = 0

    var actions = 0

    var AllTrees = 0

    for(var i = 0; i<A.length ; i++){

        AllTrees += A[i]

    }

    while(AllTrees%A.length !== 0){

        AllTrees++

    }

    var treesPerSection = AllTrees/A.length

    //console.log("trees per section:" + treesPerSection)

    for(var i = 0; i<A.length ; i++){

        //console.log(A[i])

        if (A[i]>treesPerSection){

            reserveTrees+=A[i]-treesPerSection

            A[i]-=A[i]-treesPerSection

        }

        //console.log("reserveTrees:" + reserveTrees)

        if(A[i]<treesPerSection){

            actions+=treesPerSection-A[i]

            A[i]++

        }

        /\*while(A[i]<treesPerSection){

            if(reserveTrees>0 && A[i]<treesPerSection){

                reserveTrees--

                actions++

                A[i]++

            } else if(reserveTrees !== 0 ){

                actions++

                A[i]++

            }

        }\*/

        //console.log("A[" + i + "]: " + A[i])

    }

    return actions

}