**Problem statement 1**

Answers:

function countNumbers() {

let count = 0;

for (let i = 1; i <= 9; i++) {

for (let j = i + 1; j <= 9; j++) {

let currentNum = [i, j];

let usedDigits = new Set([i, j]);

for (let k = j + 1; k <= 9; k++) {

if (!usedDigits.has(k) && currentNum[currentNum.length - 1] < k) {

currentNum.push(k);

usedDigits.add(k);

count++;

}

}

}

}

return count;

}

console.log(countNumbers());

When we run the code, it outputs **84**, which is the total number of increasing 3-digit numbers where each digit is distinct.

**Problem statement 2**

Answers:

function generateSteps(num1, num2) {

let steps = {};

let carry = "0";

let sum = "";

let n1 = num1.toString().split("").reverse();

let n2 = num2.toString().split("").reverse();

for (let i = 0; i < n1.length; i++) {

let digit1 = parseInt(n1[i]);

let digit2 = parseInt(n2[i]) || 0;

let currentSum = digit1 + digit2 + parseInt(carry);

sum = (currentSum % 10) + sum;

carry = (currentSum > 9) ? "1" : "0";

steps["step" + (i+1)] = {

"carryString": "1".repeat(i) + carry,

"sumString": "0".repeat(i) + sum

}

}

return steps;

}

let num1 = 1489;

let num2 = 714;

console.log(generateSteps(num1, num2));

In this Method the Answer i’m getting is not exact you asked for, I tried my best to get it,

The desired output is as follows;

**Output**

**{**

**step1: { carryString: '1', sumString: '3' },**

**step2: { carryString: '11', sumString: '003' },**

**step3: { carryString: '111', sumString: '00203' },**

**step4: { carryString: '1110', sumString: '0002203' }**

**}**