**Örnek Uygulama Kodu:  
JS--1**

var isSymmetric = function(root) {

// 2 kuyruklu BFS

if(!root) return true;

// diziyi kuyruk olarak kullanmanız önerilmez

let q1 = [], q2 = [];

q1.push(root.left);

q2.push(root.right);

while(q1.length > 0 && q2.length > 0){

let node1 = q1.shift();

let node2 = q2.shift();

// hepsi bitkin  
 if(!node1 && !node2) continue;

// muadili mevcut değil  
 if(!node1 || !node2) return false;

// uyumsuzluk  
 if(node1.val !== node2.val) return false;

q1.push(node1.left);

q2.push(node2.right);

q1.push(node1.right);

q2.push(node2.left);

}

// eğer her iki dizi de biterse, true değerini döndür

return q1.length === 0 && q2.length === 0;

}

**JS—2**

var isSymmetric = function(root) {

if(!root) return true;

// yardımcı işlev

const helper = (left, right) => {

// hepsi bitkin  
 if(!left && !right) return true;

// muadili mevcut değil

if(!left || !right) return false;

// uyumsuzluk

if(left.val !== right.val) return false;

return helper(left.left, right.right) && helper(left.right, right.left);

}

// yardımcı özyinelemeli işlevi çağır  
 return helper(root.left, root.right);

}

**Project Euler Kodları**

**1-**   
Problemin numarası:8

Javascript kodları:

let n = "7316717653133062491922511967442657474235534919493496983520312774506326239578318016984801869478851843858615607891129494954595017379583319528532088055111254069874715852386305071569329096329522744304355766896648950445244523161731856403098711121722383113622298934233803081353362766142828064444866452387493035890729629049156044077239071381051585930796086670172427121883998797908792274921901699720888093776657273330010533678812202354218097512545405947522435258490771167055601360483958644670632441572215539753697817977846174064955149290862569321978468622482839722413756570560574902614079729686524145351004748216637048440319989000889524345065854122758866688116427171479924442928230863465674813919123162824586178664583591245665294765456828489128831426076900422421902267105562632111110937054421750694165896040807198403850962455444362981230987879927244284909188845801561660979191338754992005240636899125607176060588611646710940507754100225698315520005593572972571636269561882670428252483600823257530420752963450";

function product(number, range){   
let newVal = 1;   
let oldVal;   
let answer = 0;   
for (var seed = 0; seed <= number.length - range; seed++) {   
oldVal = newVal;   
newVal = 1;   
for (var j = 0; j < range; j++) {   
newVal \*= number[seed + j];   
}   
if(newVal > answer){   
answer = newVal;   
}   
}   
return answer;   
}   
product(n, 13);

***Sonucu:* 23514624000**

**2-**Problemin numarası:18

Javascript kodları:

var test =["3","7 4","2 4 6","8 5 9 3"];

var triangle = ["75","95 64","17 47 82","18 35 87 10","20 04 82 47 65","19 01 23 75 03 34","88 02 77 73 07 63 67","99 65 04 28 06 16 70 92","41 41 26 56 83 40 80 70 33","41 48 72 33 47 32 37 16 94 29","53 71 44 65 25 43 91 52 97 51 14","70 11 33 28 77 73 17 78 39 68 17 57","91 71 52 38 17 14 91 43 58 50 27 29 48","63 66 04 68 89 53 67 30 73 16 69 87 40 31","04 62 98 27 23 09 70 98 73 93 38 53 60 04 23"];

function prepareNumTriangle(tri) {

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

    tri[i] = tri[i].split(" ");

    for(var j = 0; j < tri[i].length; j++) {

      tri[i][j] = parseInt(tri[i][j]);

    }

  }

}

function collapseTriangleToHighest(numTri) {

  var tmpTri = numTri;

  var botRow = tmpTri[tmpTri.length - 1];

  for(var i = 0; i < tmpTri.length-1; i++) {

    var checkRow = tmpTri[tmpTri.length - 2 - i];

    //DEBUG console.log(checkRow);

    //DEBUG console.log(botRow);

    //DEBUG console.log("");

    for(var j = 0; j < checkRow.length; j++) {

      if(botRow[j] > botRow[j + 1]) checkRow[j] += botRow[j];

      else checkRow[j] += botRow[j + 1];

    }

    botRow = checkRow;

  }

  return tmpTri[0][0];

}

prepareNumTriangle(triangle);

collapseTriangleToHighest(triangle);

***Sonucu:* 1074**

**3-**Problemin numarası:58  
  
Javascript kodları:

const createPrimes = (max) => {   
// create the primes   
const PRIMES = Array();   
const BUFFER = Array(max).fill(true);   
// loop over the buffer   
for(let i = 2; i < BUFFER.length; i++) {   
// if the buffer is true   
if(BUFFER[i]) {   
// add it into the primes   
PRIMES.push(i);   
// loop over the buffer   
for(let j = i + i; j < BUFFER.length; j += i) {   
//console.log(j);   
// set multiples false   
BUFFER[j] = false;   
}   
}   
}   
// return the primes return PRIMES;   
};   
const isPrime = (num) => num > MAX\_PRIME ? testPrime(num) : PRIME\_SET.has(num);   
const testPrime = (num) => {   
let prime = true;   
for (let i = 0; prime && i < PRIME\_ARRAY.length && PRIME\_ARRAY[i] \* PRIME\_ARRAY[i] <= num; i++) { prime = prime && !(num % PRIME\_ARRAY[i] === 0);   
}   
return prime;   
};   
const next = (diag) => {   
const next = diag[diag.length - 1] + (diag[diag.length - 1] - diag[diag.length - 2]) + 8;   
diag.push(next);   
return next;   
};   
const calc = () => {   
const diagonals = [[1, 3], [1, 5], [1, 7], [1, 9]];   
let count\_primes = 3;   
let total = 5;   
do {   
for(let i = 0; i < diagonals.length; i++) {   
total++;   
count\_primes += isPrime(next(diagonals[i])) ? 1 : 0;   
}   
} while((count\_primes / total \* 100) > 10);   
console.log((total + 1) / 2);  
 };  
 const MAX\_PRIME = 1000000;   
const PRIME\_ARRAY = createPrimes(MAX\_PRIME);   
const PRIME\_SET = new Set(PRIME\_ARRAY);   
calc();

Sonucu: **26241**

**4-**Problemin numarası:38

Javascript kodları:  
  
var panDigitals = [];

outerloop:

for (var number = 9999; number > 0; number--) {

var numString = '' + number;

var multiplier = 1;

while (numString.length < 9) {

multiplier++;

numString += (number \* multiplier);

}

if (numString.length == 9) {

for (var digit = 1; digit < 10; digit++) {

if (numString.indexOf('' + digit) == -1) {

continue outerloop;

}

}

panDigitals.push(numString);

}

}

panDigitals.sort();

console.log(panDigitals);

console.log(panDigitals[panDigitals.length - 1]);

Sonucu: **932718654**

**5-**Problemin numarası: 88

Javascript kodları:

var u = {};

işlev gen (n, r, s) {

    var k = r-s;

    eğer (k> = 2 && (! u [k] || u [k]> r)) {

        u [k] = r;

    }

    for (var i = 2; i <= n; i ++) {

        (r \* i- (s + i-1)> 12000);// koparsa

        gen (i, r \* i, s + i-1);

    }

}

for (var i = 2; i <= 12000; i ++) {

    gen (i, i, i-1);

}

var v = {};

için (var i in u) {

    v [u [i]] = doğru;

}

var w = 0;

için (var i in v) {

    w + = Sayı (i);

}

console.log ('ans:' + w);  
Sonucu: **7587457**

**LeetCode Kodları**

**1-**

Problemin numarası:38

Javascript kodları:

var countAndSay = function(n) {

let present = [1,0];

let future = [];

let pair = [0,0];

while (--n > 0) {

present.forEach(a=>{

if (a !== pair[1]) {

pair[1] ? future.push(...pair) : 0;

pair = [1, a];

} else {

pair[0]++;

};

});

future.push(0);

present = future;

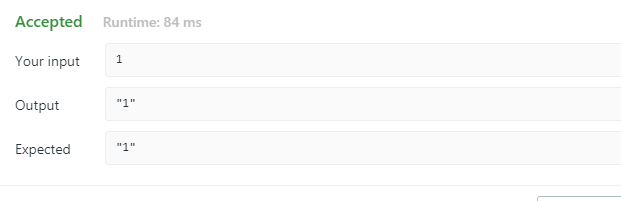
future = [];

pair = [0,0];

};

return present.slice(0, -1).join('');

};

Sonucu: ****

**2-**

Problemin numarası:58

Javascript kodları:

/\*\*

\* @param {string} s

\* @return {number}

\*/

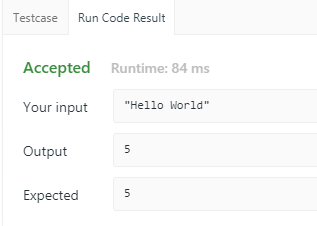
var lengthOfLastWord = function(s) {

s = s.trim();

return s.match(/\w+$/) ? s.match(/\w+$/)[0].length : 0;

};

Sonucu:



**3-**

Problemin numarası:48

Javascript kodları:

var rotate = function(matrix) {

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

for(var j=0;j<i;j++){

var temp = matrix[i][j];

matrix[i][j]=matrix[j][i];

matrix[j][i] = temp;

}

}

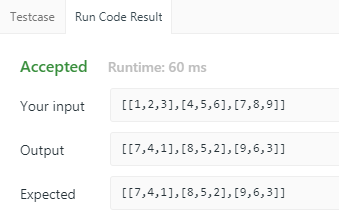
for(var i=0;i<matrix.length;i++){

matrix[i].reverse();

}

};

Sonucu:

  
**4-**Problemin numarası:18

Javascript kodları:

var fourSum = function(nums, target) {

nums.sort((a,b)=>(a-b));

const res = [];

for (let i = 0; i<nums.length-3; i++) {

if (i > 0 && nums[i] == nums[i-1]) continue;

//and below like 3-summ

for (let j = i + 1; j < nums.length - 2; j++) {

if (j > i + 1 && nums[j] == nums[j-1]) continue;

let left = j + 1;

let right = nums.length - 1;

while (left < right) {

let sum = nums[i] + nums[j] + nums[left] + nums[right];

if (sum > target) right--;

else if (sum < target) left++;

else {

res.push([nums[i], nums[j], nums[left], nums[right]]);

right--;

left++;

while(left < right && nums[left-1] == nums[left]) left++;

while(left < right && nums[right+1] == nums[right]) right--;

}

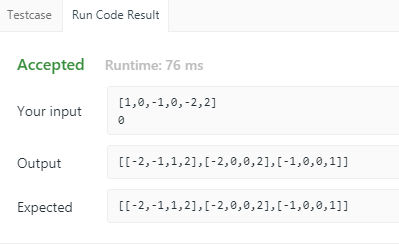
}

}

}

return res;

}  
Sonucu:



**5-**Problemin numarası:88

Javascript kodları:

var merge = function(nums1, m, nums2, n) {

if (n==0) return;

m--;

n--;

while(n>=0) {

if (nums1[m]>nums2[n]) {

nums1[m + n + 1] = nums1[m];

m--

}else {

nums1[m + n + 1]= nums2[n];

n--

}

}

};  
Sonucu:

