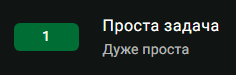
Лабораторна робота №8

NodeJS - Алгоритми



Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (input) => {

if (/^\d{2}$/.test(input)) {

const firstDigit = input.charAt(0);

const secondDigit = input.charAt(1);

console.log(`${firstDigit} ${secondDigit}`);

} else if (/^\d{1}$/.test(input)) {

const digit = input.charAt(0);

console.log(`${digit} ${digit}`);

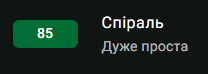
} else {

console.log('');

}

rl.close();

});



Лістинг програми:

function spiralMatrixValue(n, i, j) {

let matrix = [];

for (let k = 0; k < n; k++) {

matrix[k] = [];

}

let num = 1;

let topRow = 0, bottomRow = n - 1;

let leftCol = 0, rightCol = n - 1;

while (topRow <= bottomRow && leftCol <= rightCol) {

for (let col = leftCol; col <= rightCol; col++) {

matrix[topRow][col] = num++;

}

topRow++;

for (let row = topRow; row <= bottomRow; row++) {

matrix[row][rightCol] = num++;

}

rightCol--;

if (topRow <= bottomRow) {

for (let col = rightCol; col >= leftCol; col--) {

matrix[bottomRow][col] = num++;

}

bottomRow--;

}

if (leftCol <= rightCol) {

for (let row = bottomRow; row >= topRow; row--) {

matrix[row][leftCol] = num++;

}

leftCol++;

}

}

if (i >= 1 && i <= n && j >= 1 && j <= n) {

return matrix[i - 1][j - 1];

} else {

return undefined;

}

}

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (input) => {

const parts = input.trim().split(' ').map(Number);

const n = parts[0];

const i = parts[1];

const j = parts[2];

const result = spiralMatrixValue(n, i, j);

console.log(result);

rl.close();

});



const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

function countWords(text) {

const cleanText = text.replace(/[.,\/#!$%\^&\\*;:{}=\-\_`~()]/g, '');

const words = cleanText.split(' ');

const filteredWords = words.filter(word => word !== '');

return filteredWords.length;

}

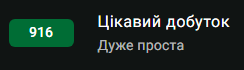
rl.question('', (input) => {

const numberOfWords = countWords(input);

console.log(`${numberOfWords}`);

rl.close();

});



Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

function countDistinctProducts(a, b, c, d) {

if (a > b) [a, b] = [b, a];

if (c > d) [c, d] = [d, c];

let products = new Set();

for (let i = a; i <= b; i++) {

for (let j = c; j <= d; j++) {

products.add(i \* j);

}

}

return products.size;

}

rl.question('', (input) => {

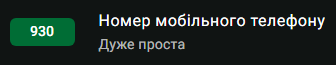
const [a, b, c, d] = input.trim().split(' ').map(Number);

const result = countDistinctProducts(a, b, c, d);

console.log(`${result}`);

rl.close();

});



Лістинг програми:  
const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (input) => {

const digitsInPhone = input.replace(/\D/g, '');

const presentDigits = new Set(digitsInPhone);

const allDigits = new Set('0123456789');

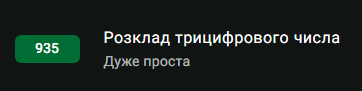
const missingDigits = [...allDigits].filter(digit => !presentDigits.has(digit));

console.log(missingDigits.length);

console.log(missingDigits.sort().join(' '));

rl.close();

});



Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (input) => {

const number = input.trim();

const isNegative = number.startsWith('-');

const absoluteNumber = isNegative ? number.slice(1) : number;

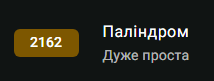
for (const digit of absoluteNumber) {

console.log(digit);

}

rl.close();

});



Лістинг програми:  
const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (input) => {

const cleanString = (str) => {

return str.toLowerCase().replace(/\s+/g, '');

};

const isPalindrome = (str) => {

const cleaned = cleanString(str);

const reversed = cleaned.split('').reverse().join('');

return cleaned === reversed;

};

if (isPalindrome(input)) {

console.log("YES");

} else {

console.log("NO");

}

rl.close();

});





Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (answer) => {

const n = parseInt(answer);

if (isNaN(n) || n <= 0 || n > 20) {

console.log("");

rl.close();

return;

}

for (let i = 0; i < n; i++) {

let row = '';

for (let j = 0; j < n; j++) {

if (j === n - 1 - i) {

row += '0';

} else if (j > n - 1 - i) {

row += '1';

} else {

row += '2';

}

}

console.log(row);

}

rl.close();

});



Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (line1) => {

rl.question('', (line2) => {

const [n, k] = line1.split(' ').map(Number);

const positions = line2.split(' ').map(Number);

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

let left = 1;

let right = positions[n - 1] - positions[0];

let result = 0;

while (left <= right) {

const mid = Math.floor((left + right) / 2);

let count = 1;

let lastPosition = positions[0];

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

if (positions[i] - lastPosition >= mid) {

count++;

lastPosition = positions[i];

}

if (count >= k) break;

}

if (count >= k) {

result = mid;

left = mid + 1;

} else {

right = mid - 1;

}

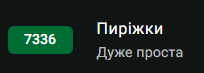
}

console.log(result);

rl.close();

});

});



Лістинг програми:

const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (answer) => {

const [a, b, n] = answer.split(' ').map(Number);

const totalKopiykas = n \* (a \* 100 + b);

const totalGrivnas = Math.floor(totalKopiykas / 100);

const finalKopiykas = totalKopiykas % 100;

console.log(`${totalGrivnas} ${finalKopiykas}`);

rl.close();

});