Министерство цифрового развития, связи и массовых коммуникаций Государственное образовательного учреждение высшего образования

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Задачи для самостоятельного решения

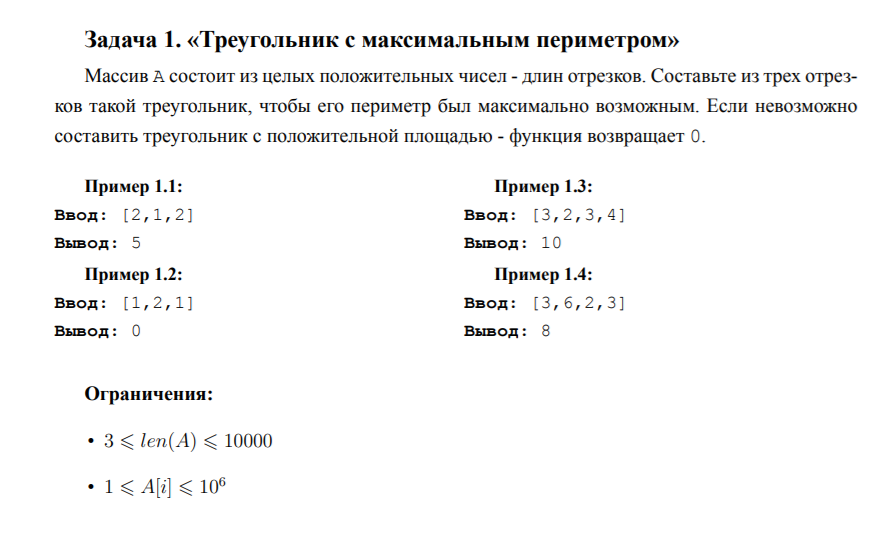
по дисциплине «Структура и алгоритмы обработки данных»

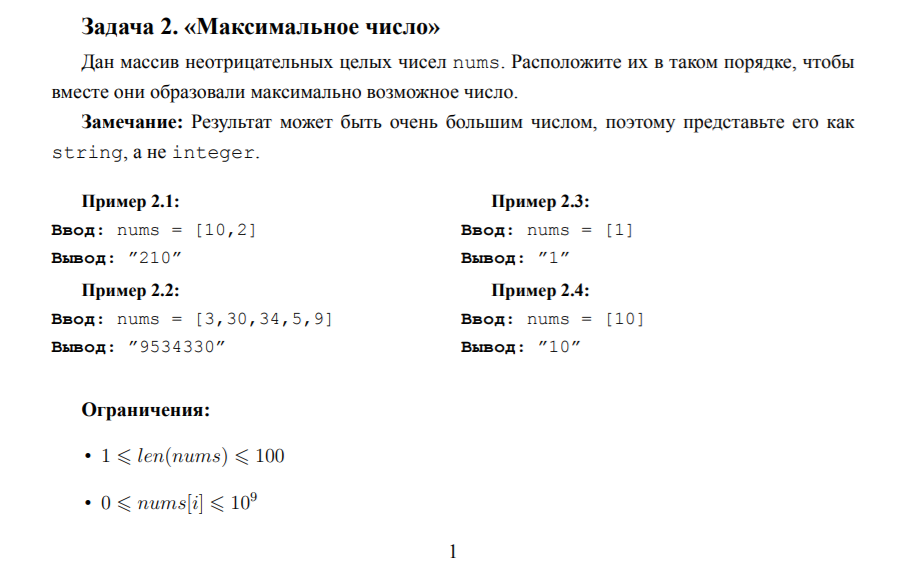
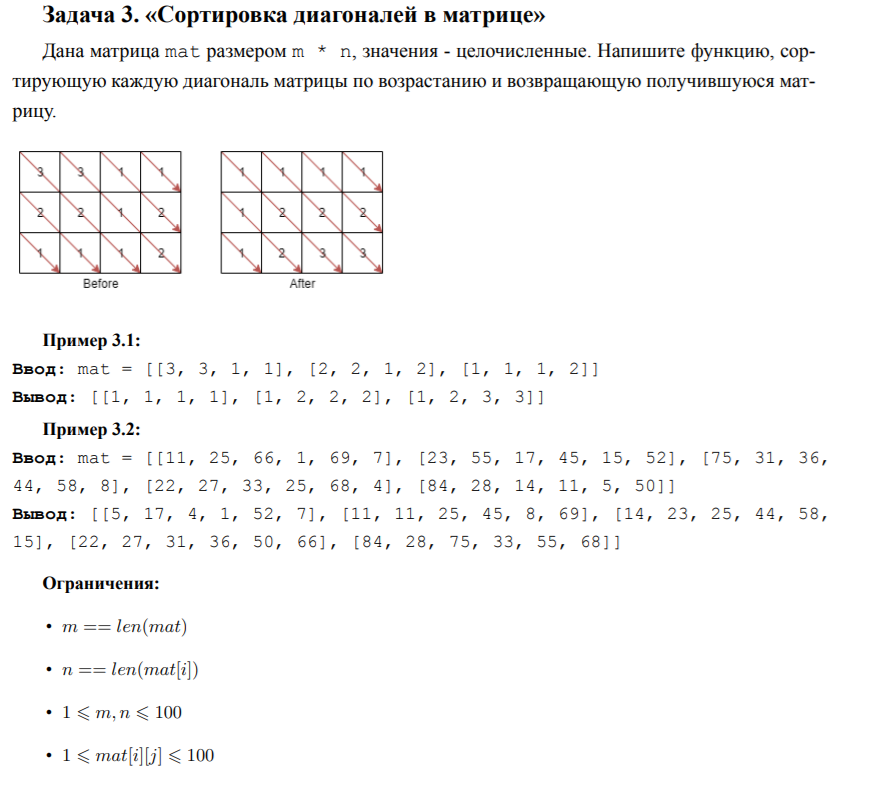
Выполнил студент группы БФИ-1901:

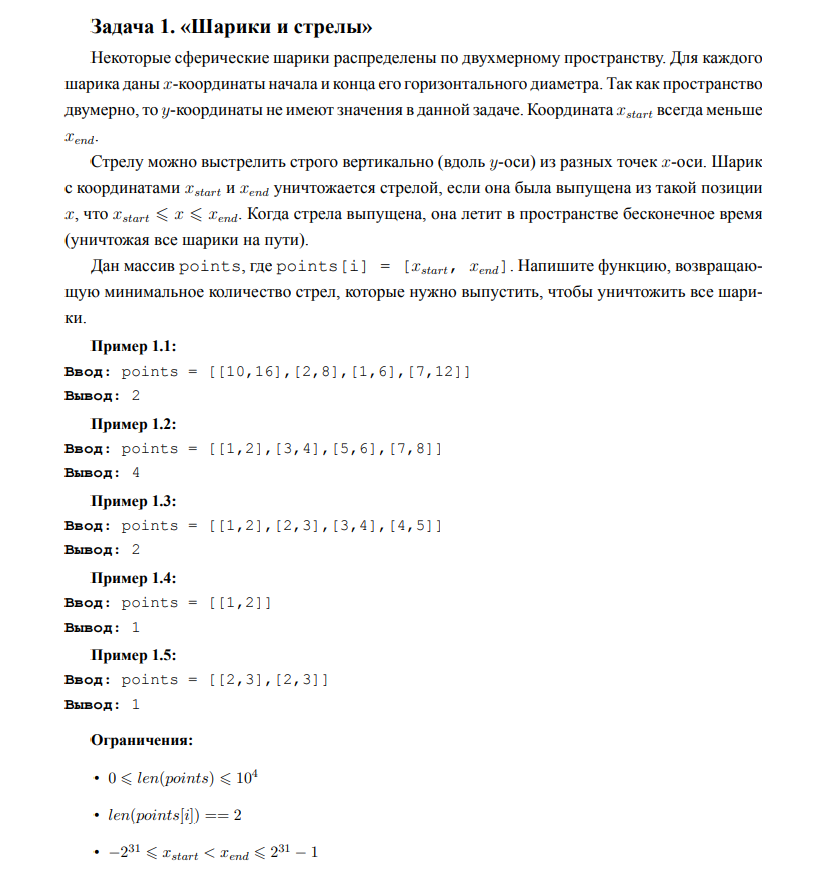
Бардюк Д.В.

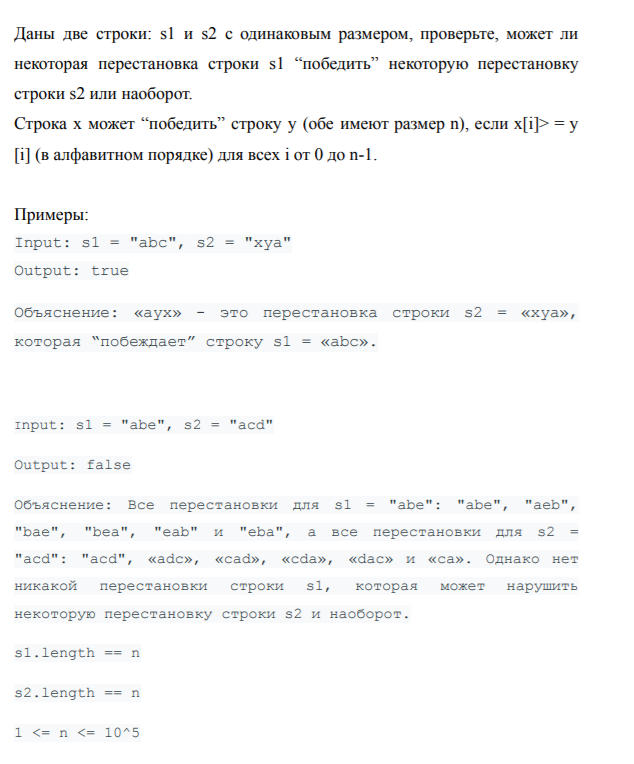
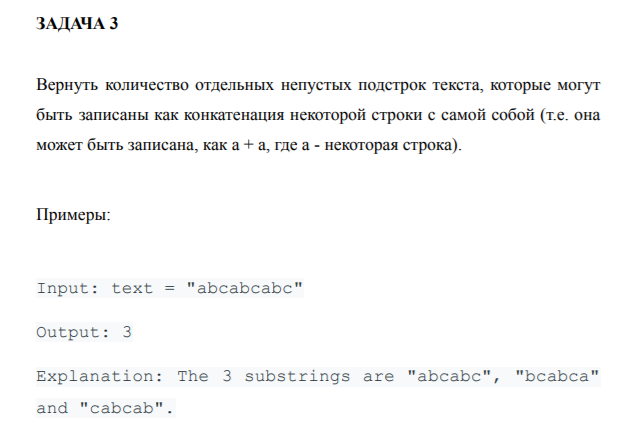
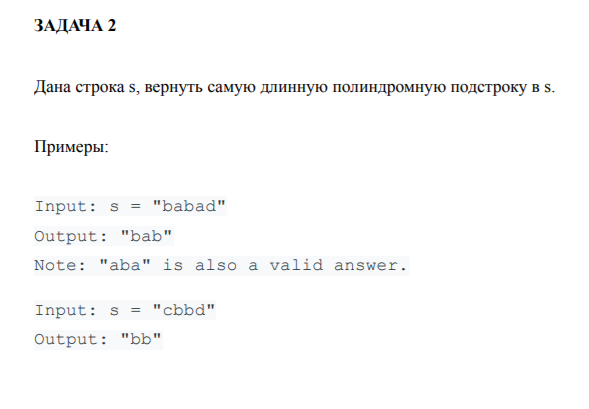
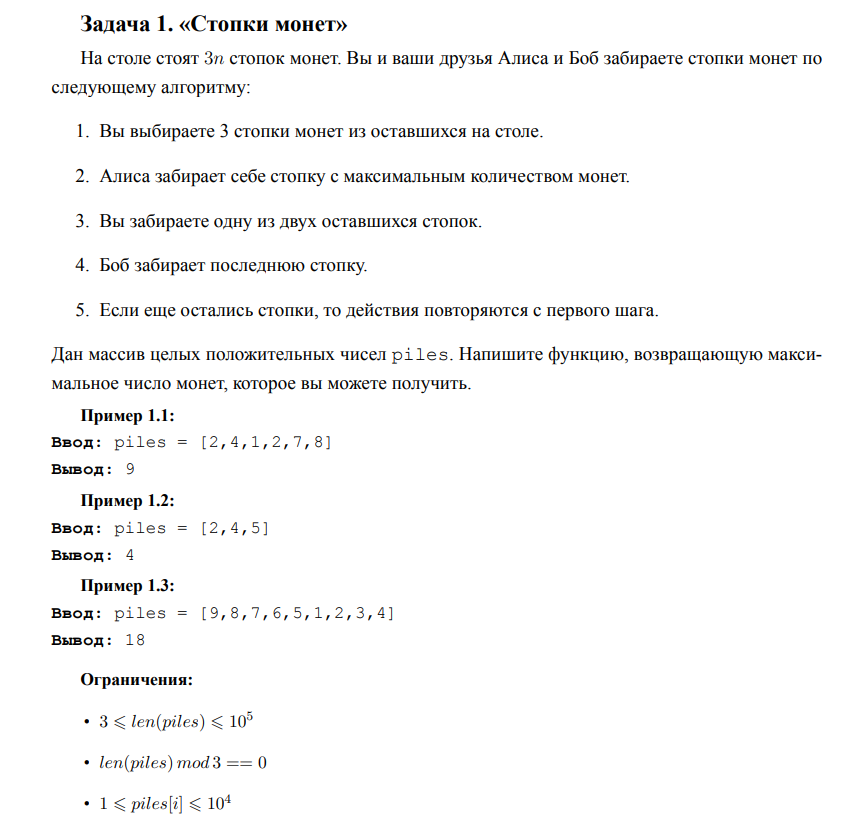
Москва 2021

# Задание





# Код программы

//Контрольные задания 1

function generateArray(lenght){

    let arr = [];

    for(let i = 0; i <lenght; i++){

        arr[i] = 0 + Math.floor(Math.random() \* (Math.pow(10,6) - 0 + 1));

    }

    return arr;

}

function thriaAngle(array){

    let per = 0, maxA, maxB,maxC;

    for (let i = 0; i < array.length - 2; i++){

        maxA = array[i];

        for (let j = 1; j < array.length - 1; j++){

            maxB = array[j];

            for (let k = 2; k < array.length ; k++){

                maxC = array[k];

                if(i != j && j != k && i != k){

                    if ((maxA + maxB > maxC) && (maxB + maxC > maxA) && (maxA + maxC > maxB)){

                        console.log(maxA , maxB, maxC);

                        if(per<maxA + maxB + maxC){

                            per = maxA + maxB + maxC;

                        }

                    }

                }

            }

        }

     }

    return per;

}

let array = generateArray(4);

console.log(array);

console.log(thriaAngle(array));

let nums = [3,30,34,5,954,1,2972,3,574,5,6,724,8,9,10,11,12,20,30,33,90,900,1000,354,99,1111,355,324]

// let nums = [0,0,0,1];

function maxNum(array){

    let check = array.every(element => element == 0);

    if(check){

        return 0;

    }

    return array

        .map((item) => item.toString())

        .sort((str1,str2) => (str2+str1) - (str1+str2))

        .join('');

}

console.log(maxNum(nums));

let mat = [[3, 3, 5, 6], [2, 2, 1, 8], [1, 1, 1, 2]]

mat = [[3, 3, 5, 6], [2, 2, 1, 8], [1, 1, 1, 2],[2, 2, 1, 8],[2, 2, 1, 8]]

mat = [[11, 25, 66, 1, 69, 7], [23, 55, 17, 45, 15, 52], [75, 31, 36, 44, 58, 8], [22, 27, 33, 25, 68, 4], [84, 28, 14, 11, 5, 50]]

mat = [[3, 3, 5, 6], [2, 2, 1, 8], [1, 1, 1, 2],[2, 2, 1, 8]]

let rowLen = mat[0].length, columnLen = mat.length

mat.forEach((e)=>{

    console.log(e);

})

console.log('sorting')

let buff = []

let triangleDia = Math.min(rowLen, columnLen ) - 1

let countMainDia = Math.abs(rowLen-columnLen)+1

let lenMainDia = Math.min(rowLen, columnLen )

function triangleUp(){

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

        //console.log(triangleDia);

        for(let j = 0; j < i+1; j++ ){

            //console.log(i,j, rowLen-1-i+j, mat[j][rowLen-1-i+j])

            buff.push(mat[j][rowLen-1-i+j])

        }

        //console.log('buff',buff );

        buff.sort((a,b)=>{return a-b})

        for(let j = 0; j < i+1; j++ ){

            //console.log(j,rowLen-1-i, i,buff[j])

            mat[j][rowLen-1-i+j] = buff[j]

        }

        buff = []

    }

}

triangleUp()

function diaMain(){

    //console.log(countMainDia);

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

        if(rowLen>columnLen){

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

                buff.push(mat[j][j+i])

            }

            buff.sort((a,b)=>{return a-b})

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

                mat[j][j+i] = buff[j]

            }

            buff = []

        }else{

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

                buff.push(mat[j+i][j])

            }

            buff.sort((a,b)=>{return a-b})

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

                mat[j+i][j] = buff[j]

            }

            buff = []

        }

    }

}

diaMain()

function triangleDown(){

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

        //console.log(triangleDia);

        for(let j = 0; j < i+1; j++ ){

            buff.push(mat[columnLen-1-i+j][j])

        }

        //console.log('buff',buff );

        buff.sort((a,b)=>{return a-b})

        for(let j = 0; j < i+1; j++ ){

            //console.log(j,rowLen-1-i, i,buff[j])

            mat[columnLen-1-i+j][j] = buff[j]

        }

        buff = []

    }

}

triangleDown()

mat.forEach((e)=>{

    console.log(e);

})

///Задание Шарики и Стрелы

function setup(){

    createCanvas(600, 400)

    frameRate(30)

}

let stages = 0

let scale = 5

let points =     [[1,2],[2,3],[3,4],[4,5],[20,30],[30,40],[40,50]]

let spectral =[]

function spectra(){

    for(let i=0; i<100; i++) { spectral.push(0) }

    points.forEach(e=>{

        let buffMax = 0

        for(let i = e[0]; i <= e[1]; i++){

            spectral[i]++

        }

    })

    console.log('Spectral: ', spectral)

    maxi()

}

function maxi(){

    console.log('-');

    let spectMax = 0

    spectral.forEach(e=>{

        (e > spectMax) ? spectMax = e : e = e

    })

    console.log(spectMax)

    shoot(spectMax)

}

function shoot(max){

    let m = spectral.indexOf(max)

    console.log('m is ',m);

    console.log(points);

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

        console.log(i,'try')

        if(points[i][0] <= m && points[i][1] >= m){

            console.log(points[i][0], points[i][1],'spliced')

            points.splice(i,1)

            i--

        }else{

            console.log(points[i][0], points[i][1],'NonSpliced')

        }

    }

    spectral = []

    if(points.length === 0){

        console.log('Success for ', stages+1, ' shots')

    }else{

        console.log(points);

        loop()

        stages++

    }

}

function drawPoints(){

    let k = 10

    points.forEach(e=>{

        let d = e[1]-e[0]

        let r = d/2

        circle((e[0]-r)\*scale+100, k\*scale, d\*scale)

        k +=10

    })

}

let x = 0

function compare(){

    console.log('+')

    line(x,0,x,600)

    x += scale

    if(x>=300){

        x = 0

        noLoop()

        spectra()

    }

}

function getArrows(){

    let l = points.length

    let last = -1

    peak.forEach(e=>{

       // if(){}

    })

}

function draw(){

    background(220)

    fill(100,200,100)

    noStroke()

    drawPoints()

    stroke(255,0,0)

    compare();

}

//Задания по строкам

function findWin(str1,str2){

    if(str1.length === 0 || str2.length === 0){

        return `it is empty`;

    }else if (str1.length !== str2.length){

        return `Can't compare it`;

    }

   let arrChar1 = str1.split('').sort();

   let arrChar2 = str2.split('').sort();

    console.log(arrChar1)

    console.log(arrChar2)

   if((arrChar1.every((el, idx) => el <= arrChar2[idx])) || (arrChar2.every((el, idx) => el <= arrChar1[idx]))) {

    return true;

   }

    return false;

}

let str1 = `abe`;

let str2 = 'acd';

console.log(findWin(str1,str2));

function isPolindrome(string){

    return string === string.split('').reverse().join('');

}

function largestPolindrome(string){

    if(string.length ===0){

        return `it is empty`;

    }

    let largest ='';

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

        for(let j = 0; j<string.length; j++){

        let substr = string.slice(i,j);

            if(isPolindrome(substr)){

                if(largest.length<substr.length){

                    largest = substr;

                }

            }

        }

    }

    if(largest.length === 0){

        return `Haven't polindrome`;

    }

    return largest;

}

let string = 'babaaabaaa';

console.log(largestPolindrome(string);

function countConcat(string){

    let count = 0;

    let map = new Map();

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

        for (let j = 1; j < string.length; j++) {

            const subString = string.slice(i, j);

            if(subString.length !== 0){

                if(string.indexOf(subString.concat(subString)) !== -1 ){

                    if(map.get(subString) !== 1){

                        map.set(subString,1);

                    }

                }

            }

        }

    }

console.log(map);

    return count;

}

console.log(countConcat('abcabcabc'));

//Задача с монетами  
const {DeQueue} = require(`../4lab/queue`)

let array = [9,8,7,6,5,1,2,3,4];

let myCoins = 0;

function coins(array){

    let deq = new DeQueue();

    array.sort().map(item => deq.pushFront(item));

    if (array.length % 3 !== 0){

        return `can't do it`;

    }

        else{

            while(!deq.isEmpty()){

                deq.popFront();

                myCoins+=deq.popFront();

                deq.popBack();

            }

        }

    return myCoins;

}

console.log(`Наша наибольшая сумма: ${coins(array)}`);

**Результат Работы**

На рисунке 1 представлен результат работы программы составления наибольшего числа



Рисунок - 1 Результат работы программы составления наибольшего числа

На рисунке 2 представлен результат работы программы поиска палиндрома



Рисунок - 2 Результат работы программы поиску палиндрома

На рисунке 3 представлен результат работы программы поиска количества конкатенаций в строке

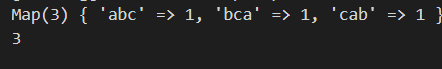


Рисунок - 3 Результат работы программы поиска количеств конкатенаций в строке

На рисунке 4 представлен результат работы программы по поиску наибольшей суммы монет в игре из задачи «Стопки монет»



Рисунок - 4 Результат работы программы проверки голосований

**Вывод:** в ходе выполнения данных заданий я применил на практике знания полученные в течении курса, разработал программы для решения заданий. Внедрение полученного за курс материала помогло оптимизировать мои алгоритмы.