МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Рекурсия и головоломки

ОТЧЕТ

по лабораторной работе № 4

по дисциплине

Технологии программирования \_

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Капранов С.Н.. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антонова С.И.

(подпись) (фамилия, и.,о.)

18-ИСТ-2\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

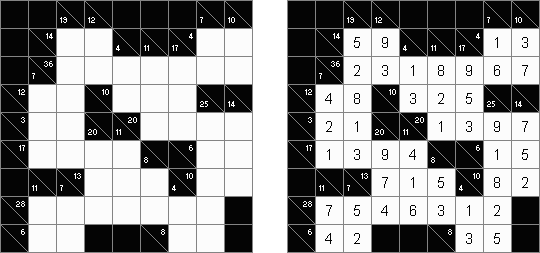
Нижний Новгород  
2019

Содержание

[**Текст задания** 3](#_Toc23886006)

# **Текст задания**

**Задача 1. Какуро** (*"Kakuro"*) - это числовая головоломка, математический эквивалент кроссворда. Необходимо вставить в клетки цифры от 1 до 9, причем некоторые клетки неактивны (такие клетки помечены черным цветом). В клетках с заданными числовыми значениями правое верхнее значение означает сумму цифр в ряду, а значение снизу слева равно сумме цифр столбца ниже клетки. Например, число 6 можно представить как сумму 1 и 5, 2 и 4; одинаковые цифры (3 и 3) использовать запрещено.



**Текст программы с комментариями**

#include <iostream>

#include <fstream>

using namespace std;

int const sizeField = 10;

int const c\_max = 9;

struct cell

{

int down, right; //сумма по вертикали и горизонтали

int val = -1; //ячейка, которую нужно заполнить

int q\_down = 0, q\_right = 0; //количество ячеек в бдоке

int pos\_down[c\_max] = {1,2,3,4,5,6,7,8,9}; //возможные значения по вертикали

int pos\_right[c\_max] = {1,2,3,4,5,6,7,8,9}; //возможные значения по горизонтади

};

cell field[sizeField][sizeField];

//=================================================================================

//копирует значения полей

//arr1 - исходное

//arr2 - копия

void Copy(cell arr1[sizeField][sizeField], cell arr2[sizeField][sizeField]);

//=================================================================================

//решает задачу рекрсивным перебором

//возвращает TRUE при верной подстановке

//a - поле

bool Rec(cell a[sizeField][sizeField]);

//=================================================================================

int main()

{

setlocale(LC\_ALL, "Rus"); //подключаем русскую локализацию

int q; // для подсчета количества ячеек в блоке

int i, j; //счетчики

//считываем информацию об исходном поле

ifstream fin;

fin.open("in.txt");

if (fin.is\_open())

{

for(i=0; i<sizeField; i++)

{

for(j=0; j<sizeField; j++)

{

fin>>field[i][j].down;

}

}

for(i=0; i<sizeField; i++)

{

for(j=0; j<sizeField; j++)

{

fin>>field[i][j].right;

if(field[i][j].right==0)

{

field[i][j].val = 0;

}

}

}

}

else cout<<"Not founded"<<endl;

fin.close();

//подсчет количества ячеек в блоке

for(i=0; i<sizeField; i++)

{

for(j=0; j<sizeField; j++)

{

q=1;

if(field[i][j].down>0)

{

while(field[i+q][j].val==0) q++;

field[i][j].q\_down = q-1;

}

}

}

for(i=0; i<sizeField; i++)

{

for(j=0; j<sizeField; j++)

{

q=1;

if(field[i][j].right>0)

{

while(field[i][j+q].val==0) q++;

field[i][j].q\_right = q-1;

}

}

}

for(i=0; i<sizeField; i++)

{

cout<<endl;

for(j=0; j<sizeField; j++)

{

cout.width(2);

if(field[i][j].val!=-1) cout<<field[i][j].val;

else cout<<'.';

}

}

Rec(field);//вызов функции, решающей задачу пребором

//вывод результата

cout<<endl;

for(int i=0; i<sizeField; i++)

{

cout<<endl;

for(int j=0; j<sizeField; j++)

{

cout.width(2);

if(field[i][j].val!=-1) cout<<field[i][j].val;

else cout<<'.';

}

}

}

//====================================================================================================================

void Copy(cell arr1[sizeField][sizeField], cell arr2[sizeField][sizeField])

{

int i, j, t;//счетчики

for(i=0; i<sizeField; i++)

{

for(j=0; j<sizeField; j++)

{

arr2[i][j].down = arr1[i][j].down;

arr2[i][j].right = arr1[i][j].right;

arr2[i][j].val = arr1[i][j].val;

arr2[i][j].q\_down = arr1[i][j].q\_down;

arr2[i][j].q\_right = arr1[i][j].q\_right;

for(t=0; t<c\_max; t++)

{

arr2[i][j].pos\_down[t] = arr1[i][j].pos\_down[t];

arr2[i][j].pos\_right[t] = arr1[i][j].pos\_right[t];

}

}

}

}

//====================================================================================================================

bool Rec(cell a[sizeField][sizeField])

{

int d, r;

int i, j, t; //счетчики

int sum\_d, sum\_r; //сумма заполненных ячеек по вертикали и горизонтали

cell a2[sizeField][sizeField]; //массов для записи выбранного значения

for(int i=0; i<sizeField; i++)

{

for(int j=0; j<sizeField; j++)

{

if(a[i][j].val==0)//если встретилась пустая клетка

{

for(t=0; t<c\_max; t++)

{

if((a[i][j].pos\_down[t]==a[i][j].pos\_down[t])&&(a[i][j].pos\_down[t]!=0))

{

Copy(a,a2);

a2[i][j].val = a2[i][j].pos\_down[t];//записываем выбранное значение

//подсчет суммы ячеек в блоках

d=0;sum\_d = 0;

while(a2[i-d][j].val!=-1)

{

sum\_d = sum\_d + a2[i-d][j].val;

d++;

}

r=0;sum\_r = 0;

while(a2[i][j-r].val!=-1)

{

sum\_r = sum\_r + a2[i][j-r].val;

r++;

}

//если оба блока заполнены

if(((sum\_d==a2[i-d][j].down)&&((d+1)==a2[i-d][j].q\_down))&&((sum\_r==a2[i][j-r].right)&&((r+1)==a2[i][j-r].q\_right)))

{

if(Rec(a2)==1) return 1;

}

//если сумма не превышает необходимую

else if((sum\_d<=a2[i-d][j].down)&&(sum\_r<=a2[i][j-r].right))

{

if(sum\_d!=a2[i-d][j].down)

{

d=0;

while(a2[i+d][j].val!=-1)

{

a2[i+d][j].pos\_down[t] = 0;//удаляем выбранное значения из возможных для всех ячеек в блоке

d++;

}

}

if(sum\_r!=a2[i][j-r].right)

{

r=0;

while(a2[i][j+r].val!=-1)

{

a2[i][j+r].pos\_down[t] = 0;

r++;

}

}

if((d!=1)&&(r!=1))//учитываем ситуацию, если блок заполнен, а сумма не достигнута

{

if(Rec(a2)==1)

{

field[i][j].val = a[i][j].pos\_down[t];

return 1;

}

}

}

else return 0;

}

}

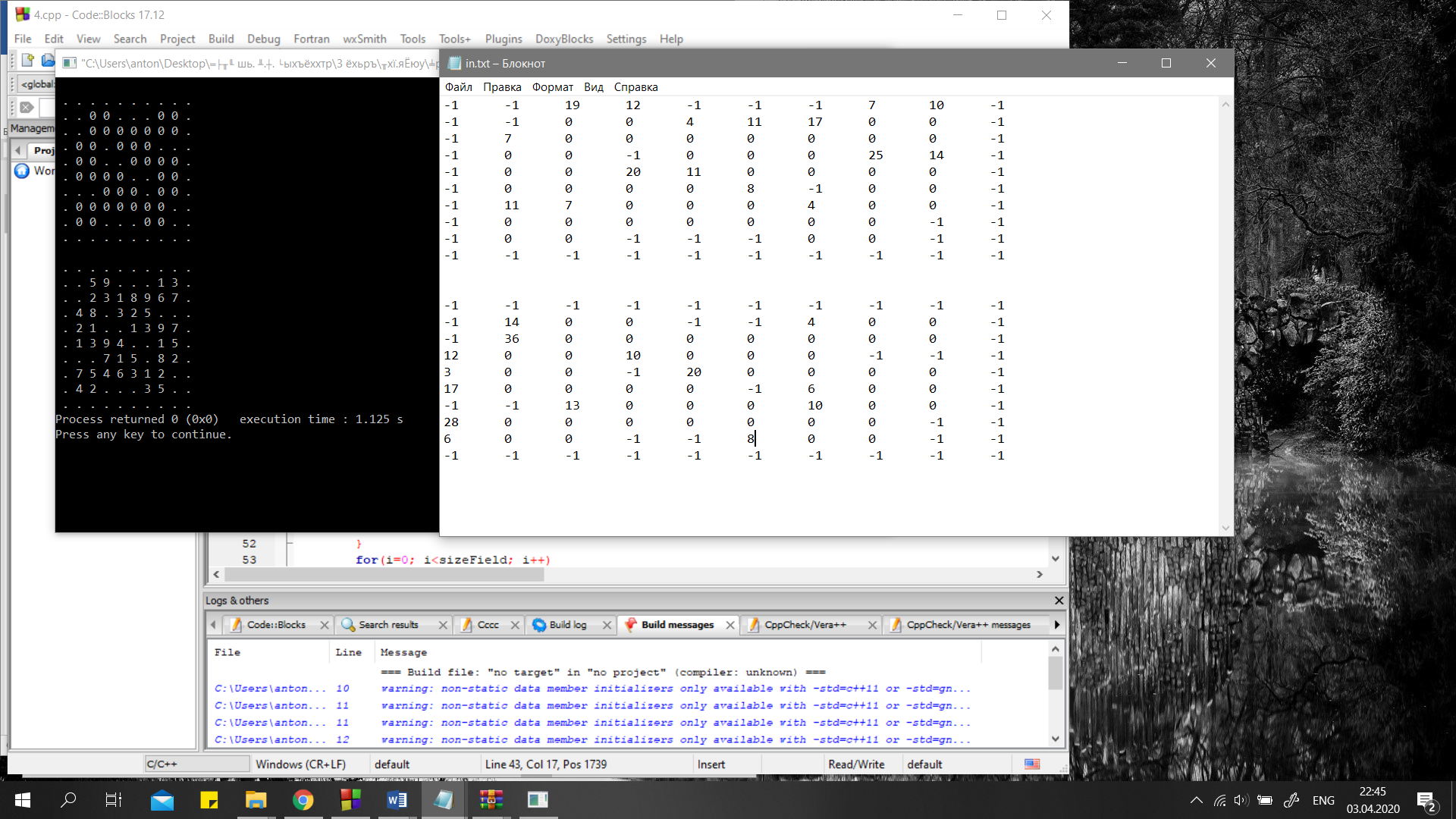
return 0;

}

}

}

}

**Принтскрины экранов с входными и выходными данными.**