Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа  
“8 ферзей”**

Выполнил:   
студент группы РИС-23-1б   
Сингур Иван Сергеевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь, 2024 г.

**Условие:** «Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого».

Подразумевается, что ферзь бьёт все клетки, расположенные по вертикалям, горизонталям и обеим диагоналям.

**Анализ задачи:**

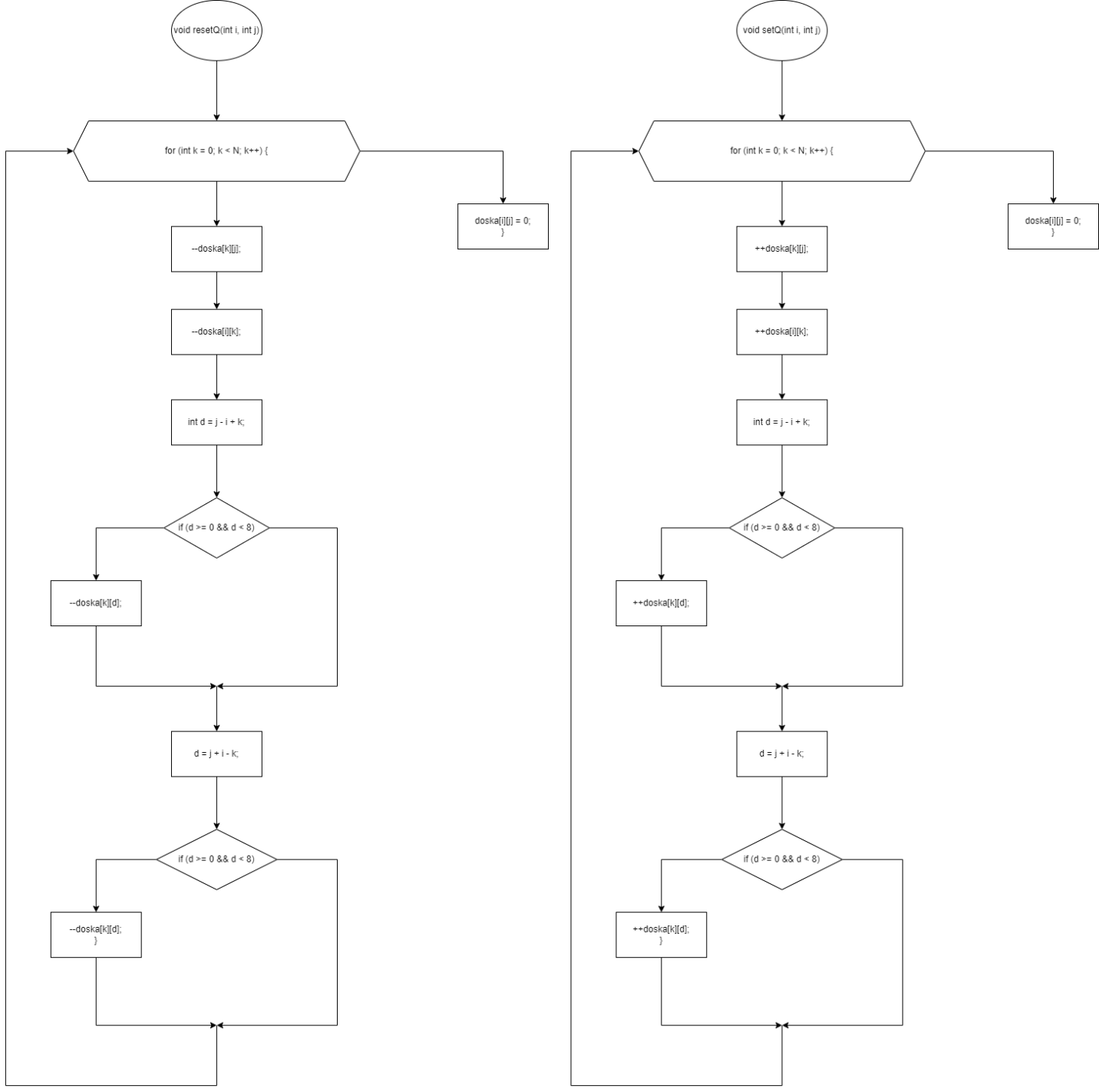
1)На каждом шаге рекурсии программа пытается расставить ферзя на очередной строке доски, проверяя при этом, не находится ли данный ферзь под атакой других ферзей. (tryQ)

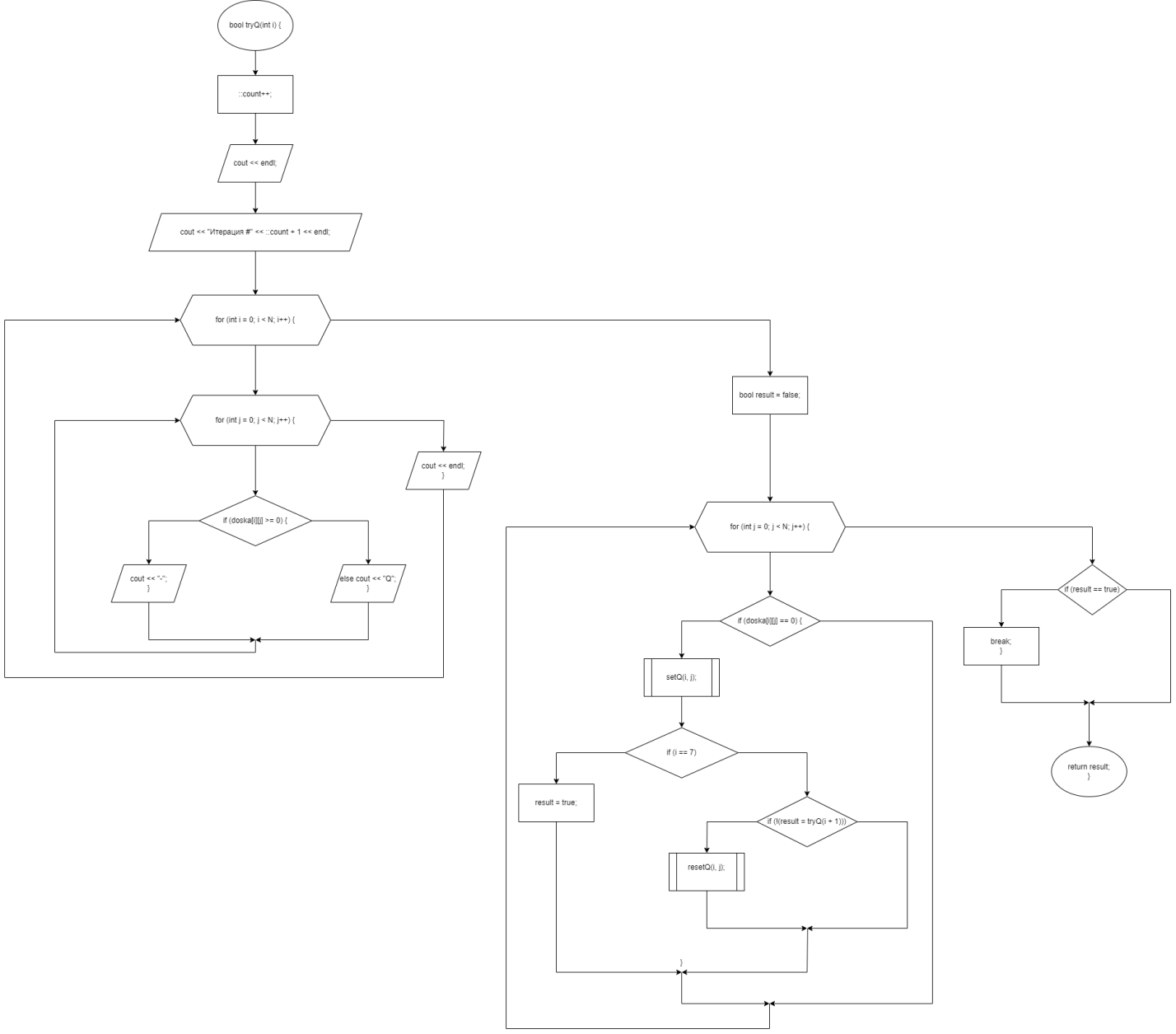
2)Если ферзь может быть установлен на данной позиции, программа переходит к следующей строке, повторяя процесс до тех пор, пока все 8 ферзей не будут успешно установлены на доске. (setQ)

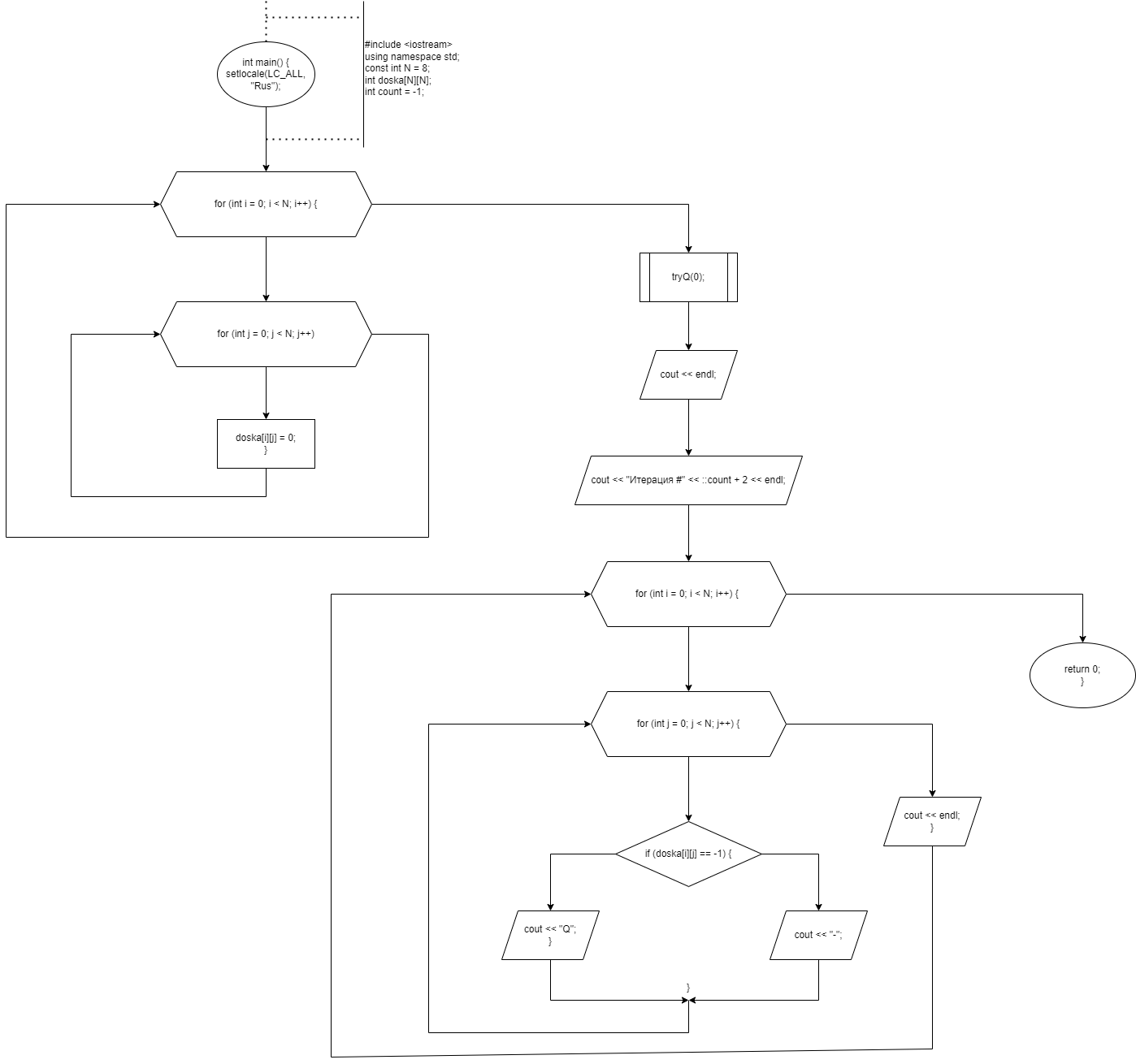
3)В случае, если на каком-то шаге программа не сможет расставить ферзя безопасным образом, она отменит ход и попробует другой вариант. Программа будет продолжать перебор вариантов, пока не найдет корректное решение задачи или не исчерпает все возможные комбинации. (resetQ)

4)После завершения выполнения программы, на экран будет выведено правильное решение задачи - расстановка 8 ферзей на шахматной доске таким образом, чтобы они не угрожали друг другу.

**Блок-схема:**

****

****

****

**Код на языке C++:**

#include <iostream>

using namespace std;

const int N = 8;

int doska[N][N]; //Создаем массив doska размером 8x8, который представляет шахматную доску.

int count = -1; //Создаём счетчик, который будет считать кол-во итераций

void resetQ(int i, int j) { //Функция resetQ, которая убирает ферзей, если последующий ход невозможен

for (int k = 0; k < N; k++) {

--doska[k][j];

--doska[i][k];

int d = j - i + k;

if (d >= 0 && d < 8)

--doska[k][d];

d = j + i - k;

if (d >= 0 && d < 8)

--doska[k][d];

}

doska[i][j] = 0;

}

void setQ(int i, int j) { //Функция setQ, которая занимается расстановкой ферзей

for (int k = 0; k < N; k++) {

++doska[k][j];

++doska[i][k];

int d = j - i + k;

if (d >= 0 && d < 8)

++doska[k][d];

d = j + i - k;

if (d >= 0 && d < 8)

++doska[k][d];

}

doska[i][j] = -1;

}

bool tryQ(int i) { //Создаем функцию tryQ(), которая рекурсивно пытается расставить ферзей на шахматной доске.

::count++;

cout << endl;

cout << "Итерация #" << ::count + 1 << endl;

for (int i = 0; i < N; i++) { //Вывод каждой итерации на экран

for (int j = 0; j < N; j++) {

if (doska[i][j] >= 0) {

cout << "-";

}

else cout << "Q";

}

cout << endl;

}

bool result = false;

for (int j = 0; j < N; j++) {

if (doska[i][j] == 0) {

setQ(i, j);

if (i == 7)

result = true;

else

if (!(result = tryQ(i + 1)))

resetQ(i, j);

}

if (result == true)

break;

}

return result;

}

int main() {

setlocale(LC\_ALL, "Rus");

for (int i = 0; i < N; i++) {

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

doska[i][j] = 0;

} // Заполняем шахмотную доску 0

tryQ(0);

cout << endl;

cout << "Итерация #" << ::count + 2 << endl;

for (int i = 0; i < N; i++) { //Вывод правильного решения на экран

for (int j = 0; j < N; j++) {

if (doska[i][j] == -1) {

cout << "Q";

}

else cout << "-";

}

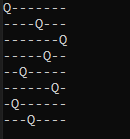
cout << endl;

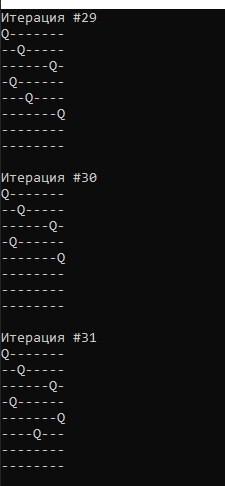
}

return 0;

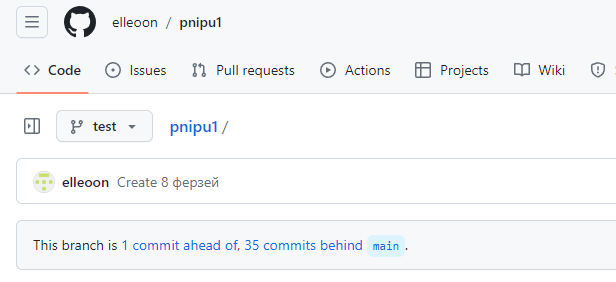
}

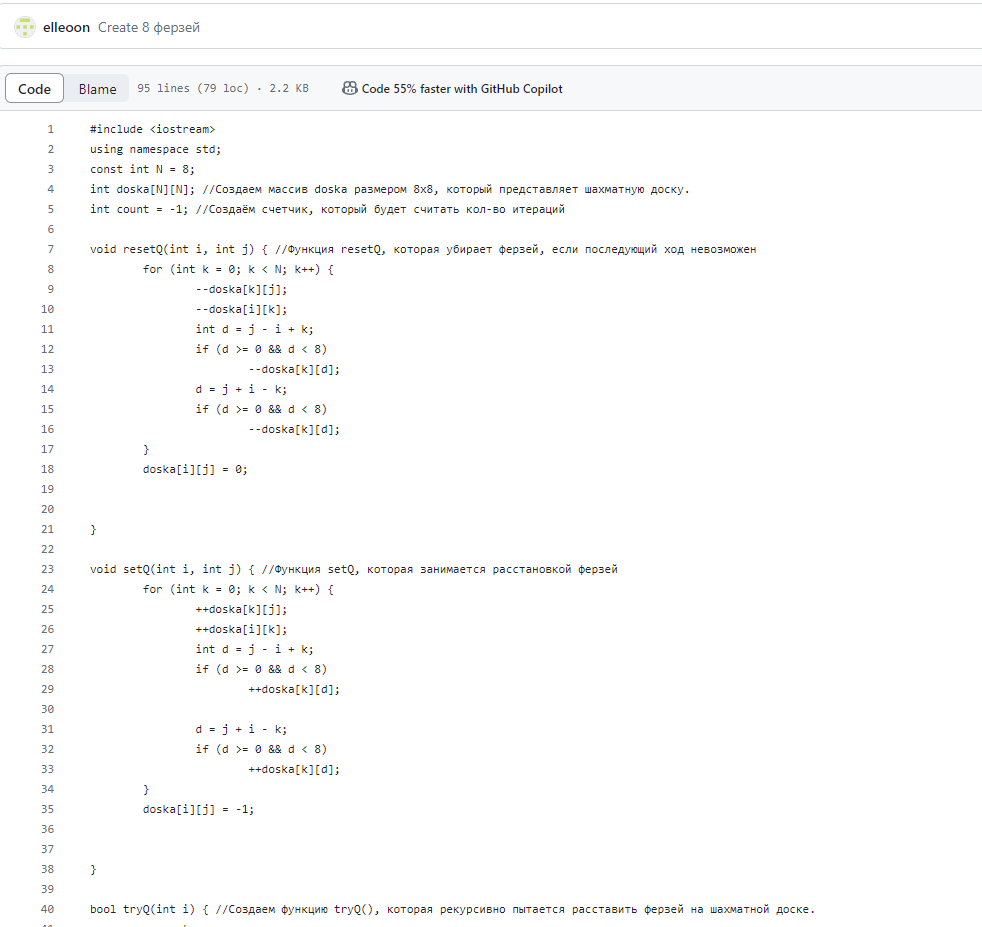
**Работа программы:**





**Cкрины из гита:**





**Ссылка на гит:**

https://github.com/elleoon/pnipu1

**Вывод:** Задача была выполнена. Всё получилось.