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

ПНИПУ

**Лабораторная работа  
"Задача о 8 ферзях"**

Выполнил:   
студент группы РИС-23-1б   
Карпов Иван Васильевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь 2024 г.

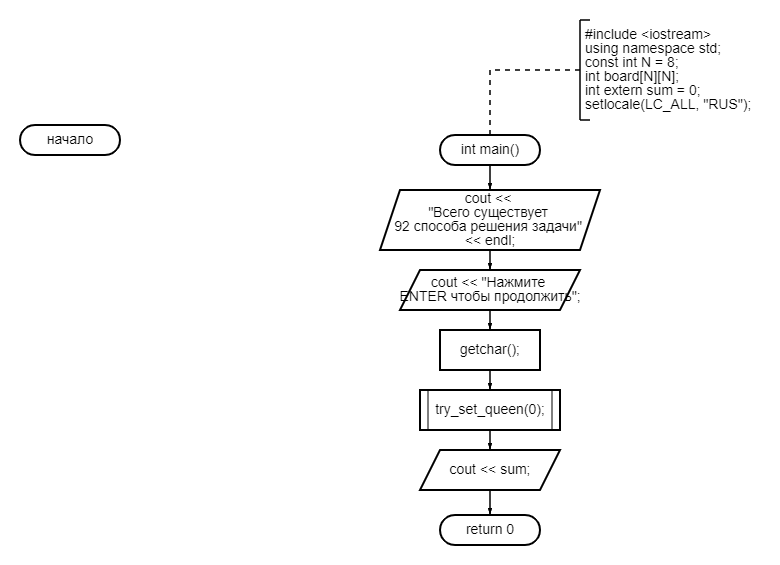
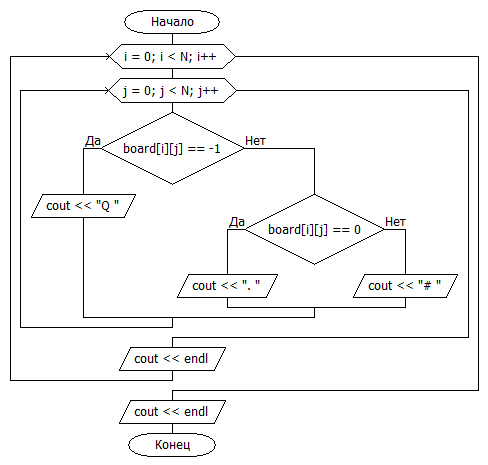
**Условие:**

Расположить 8 ферзей на шахматной доске так, чтобы они не били друг друга

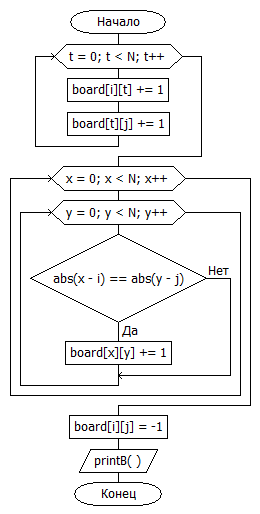
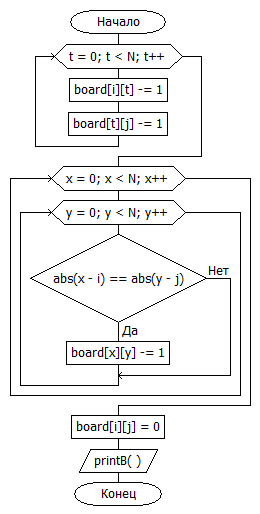
**Анализ:**

1. Будем использовать рекурсию
2. За установку и удаления ферзя будут отвечать отдельные фукции
3. Изначально дочка заполнена 0
4. Функция setQ принимает координаты ферзя, отмечает все его возможные ходы (прибавляя 1 в ячейки матрицы) и ставит самого ферзя (ячейка = -1)
5. Функция delQ принимает координаты ферзя, удаляет все его возможные ходы (отнимая 1 из ячейки матрицы) и убирает самого ферзя (ячейка = 0)
6. Функция try\_set\_queen принимает индекс строки, на которую попытается поставить ферзя, проходится по столбцам строки и, если находит свободное место (в которое еще не ставили ферзя), то ставит ферзя и вызывает себя со значением на 1 больше.
7. Иначе, если место не найдено, она вызывает саму себя со значением на 1 меньше
8. На каждом шаге выводится шахматная доска.
9. Как только достигается 8 (последняя) строка счетчик решений увеличивается и начинается поиск следующего решения.

**Блок схема:**

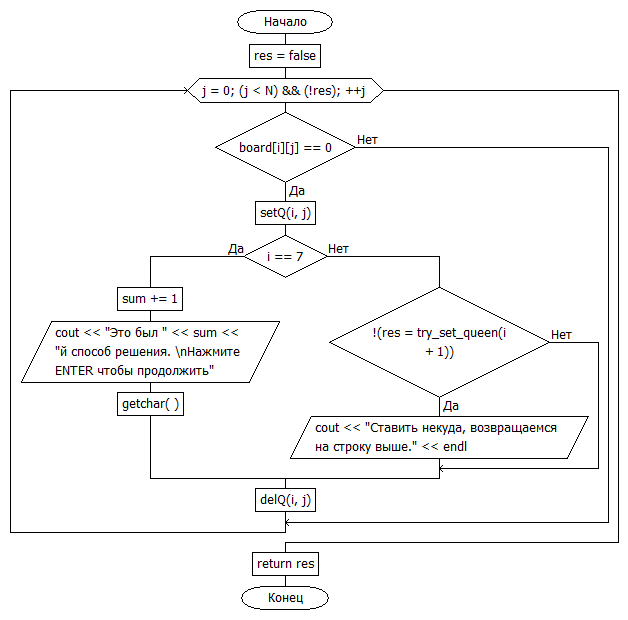
****

printB();

****

delQ();

setQ();

****

try\_set\_queen();

**Код:**

#include <iostream>

using namespace std;

const int N = 8;

int board[N][N];

int extern sum = 0;

void printB()

{

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

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

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

cout << "Q ";

}

else if (board[i][j] == 0) {

cout << ". ";

}

else

{

cout << "# ";

}

}

cout << endl;

}

cout << endl;

}

void setQ(int i, int j)

{

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

{

board[i][t] += 1;

board[t][j] += 1;

}

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

{

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

{

if (abs(x - i) == abs(y - j))

{

board[x][y] += 1;

}

}

}

board[i][j] = -1;

printB();

}

void delQ(int i, int j)

{

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

{

board[i][t] -= 1;

board[t][j] -= 1;

}

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

{

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

{

if (abs(x - i) == abs(y - j))

{

board[x][y] -= 1;

}

}

}

board[i][j] = 0;

printB();

}

bool try\_set\_queen(int i)

{

bool res = false;

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

if (board[i][j] == 0)

{

setQ(i, j);

if (i == 7)

{

sum += 1;

cout << "Это был " << sum << "й способ решения. \nНажмите ENTER чтобы продолжить";

getchar();

}

else

if (!(res = try\_set\_queen(i + 1)))

cout << "Ставить некуда, возвращаемся на строку выше." << endl;

delQ(i, j);

}

}

return res;

}

int main()

{

setlocale(LC\_ALL, "RUS");

cout << "Всего существует 92 способа решения задачи" << endl;

cout << "Нажмите ENTER чтобы продолжить";

getchar();

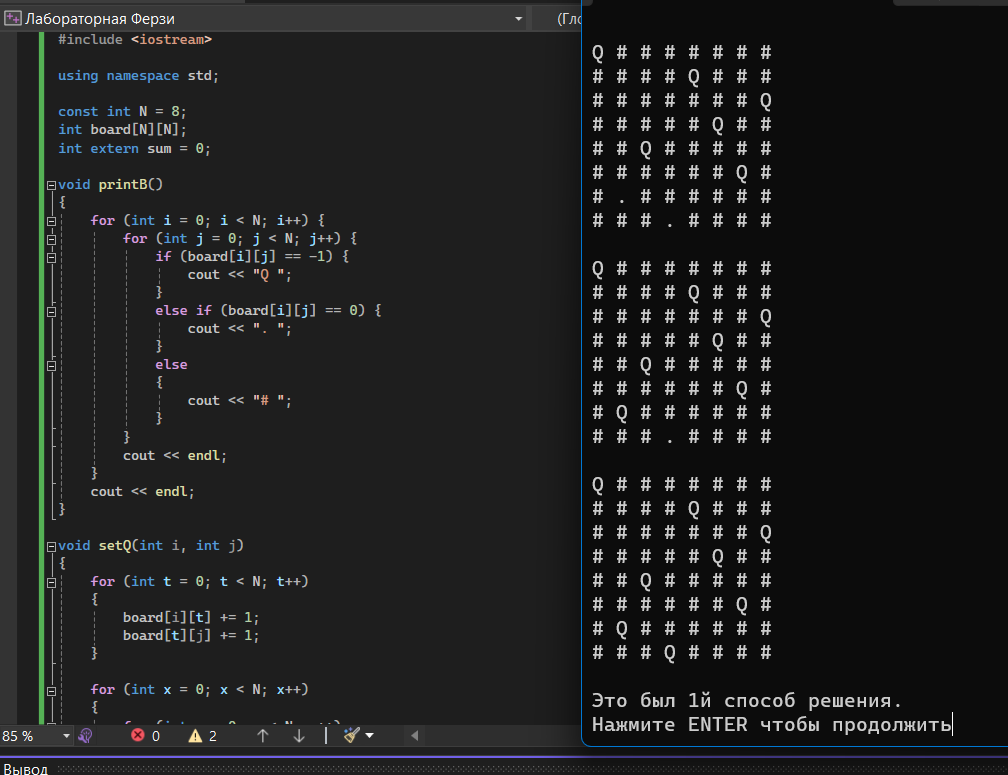
try\_set\_queen(0);

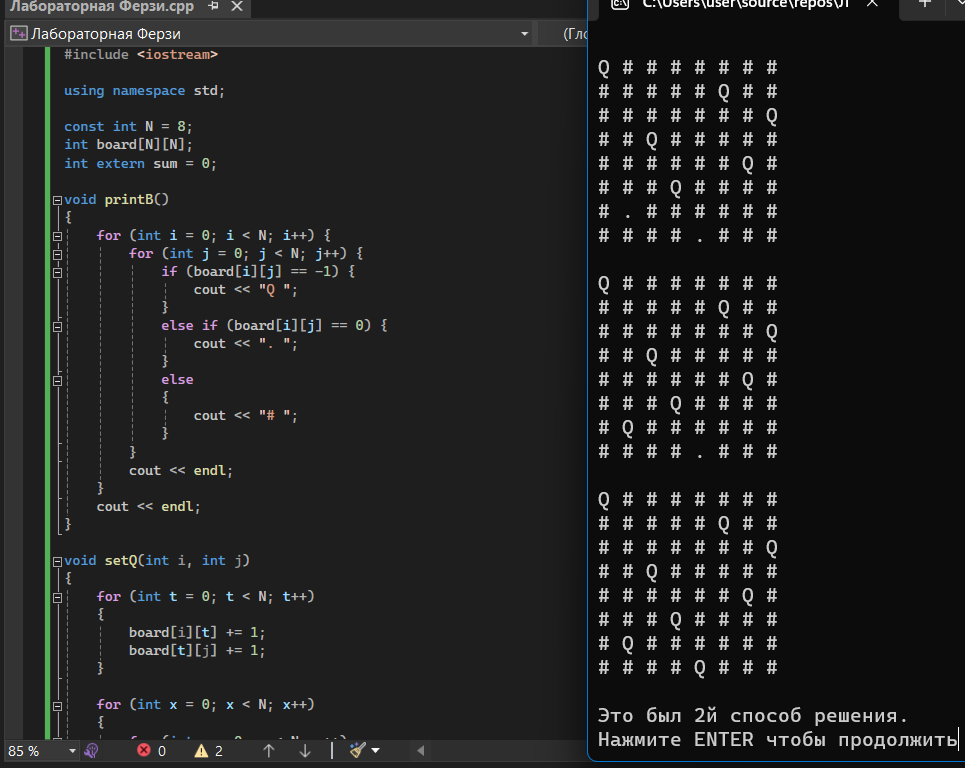
cout << sum;

return 0;

}

**Работа программы:**

****

****