

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Пермский национальный исследовательский политехнический  
университет»**

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»  
направление подготовки: 09.03.01– «Информатика и вычислительная  
техника»

**Лабораторная работа  
по дисциплине «Информатика»  
на тему «Задача о 8 ферзях»**

Выполнил студент гр. ИВТ-23-16

Бакин Владислав Артемович

Проверил:

доц. каф. ИТАС

Полякова Ольга Андреевна

\_\_\_\_\_

\_\_\_\_\_  
(оценка)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

г. Пермь, 2023

## Содержание

## Постановка задачи

Задача о восьми ферзях — широко известная комбинаторная задача по расстановке фигур на шахматной доске. Исходная формулировка: «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.

## Особенности решения

Задачу нужно решить, с использованием рекурсии на языке C++.

Общее число возможных расположений 8 ферзей на 64-клеточной доске равно  $4426165368 = 64! / (8! * (64 - 8)!)$  (формула сочетаний). Общее число возможных расположений, удовлетворяющих условию задачи, равно 92.

## Решение

### 1 Блок-схема программы

Блок-схема, на которой изображены директивы препроцессора, пространство имен, глобальный двумерный массив `int board[8][8]`, прототипы функций и функция `int main()`, показана на рисунке 1.

Блок-схема, на которой изображена функция `void setQueen(int i, int j)`, показана на рисунке 2.

Блок-схема, на которой изображена функция `void removeQueen(int i, int j)`, показана на рисунке 3.

Блок-схема, на которой изображена функция `void printPosition()`, показана на рисунках 4 и 5.

Блок-схема, на которой изображена функция `void solve(int i)`, показана на рисунке 6.

[ОВ]

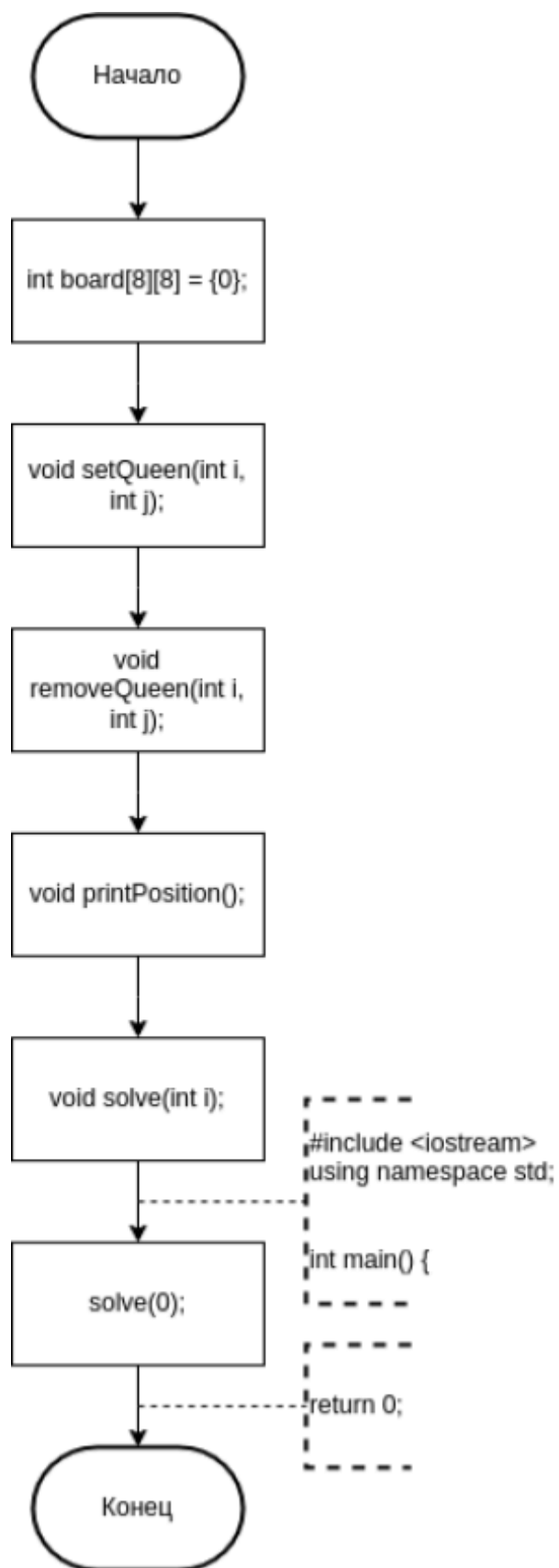


Рисунок 1

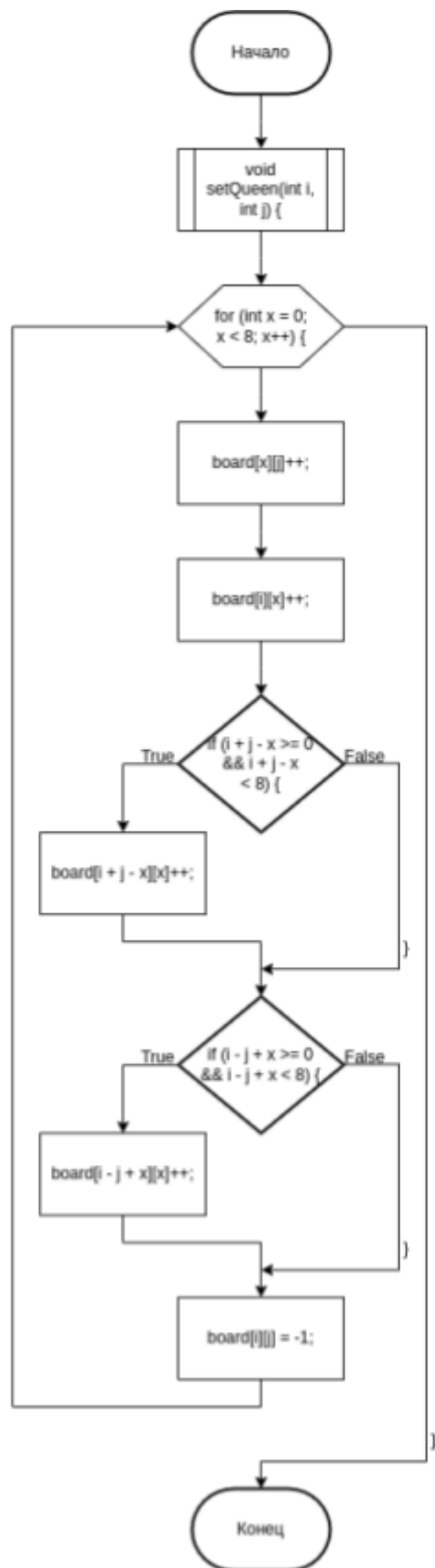


Рисунок 2

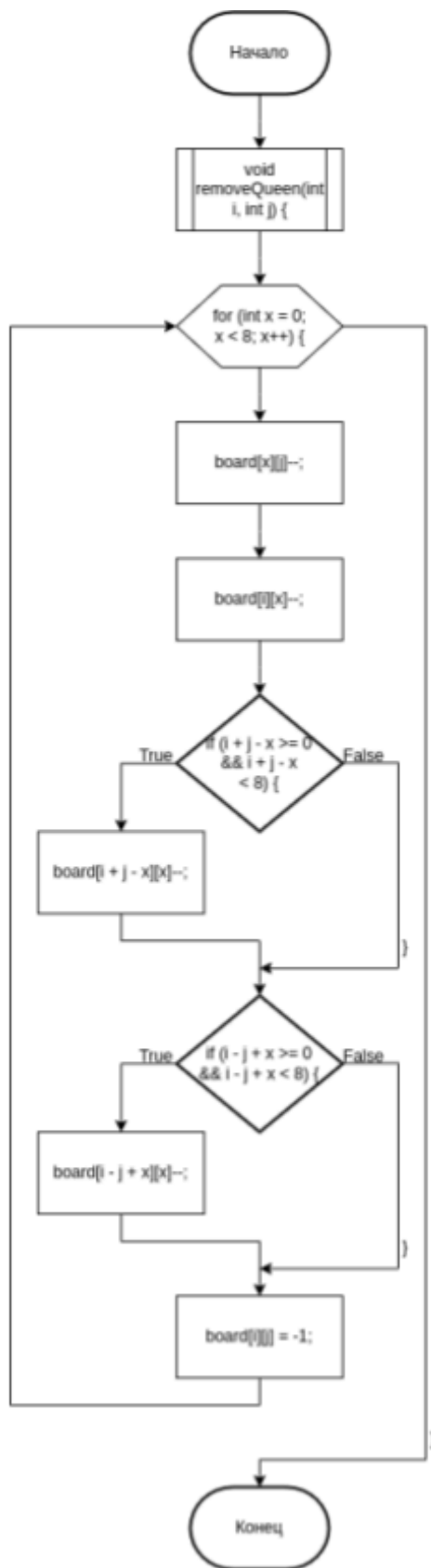


Рисунок 3

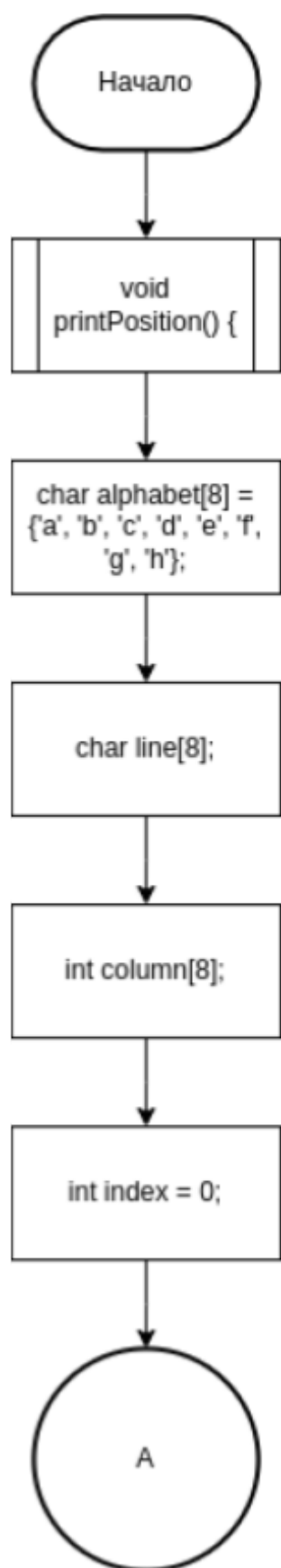


Рисунок 4

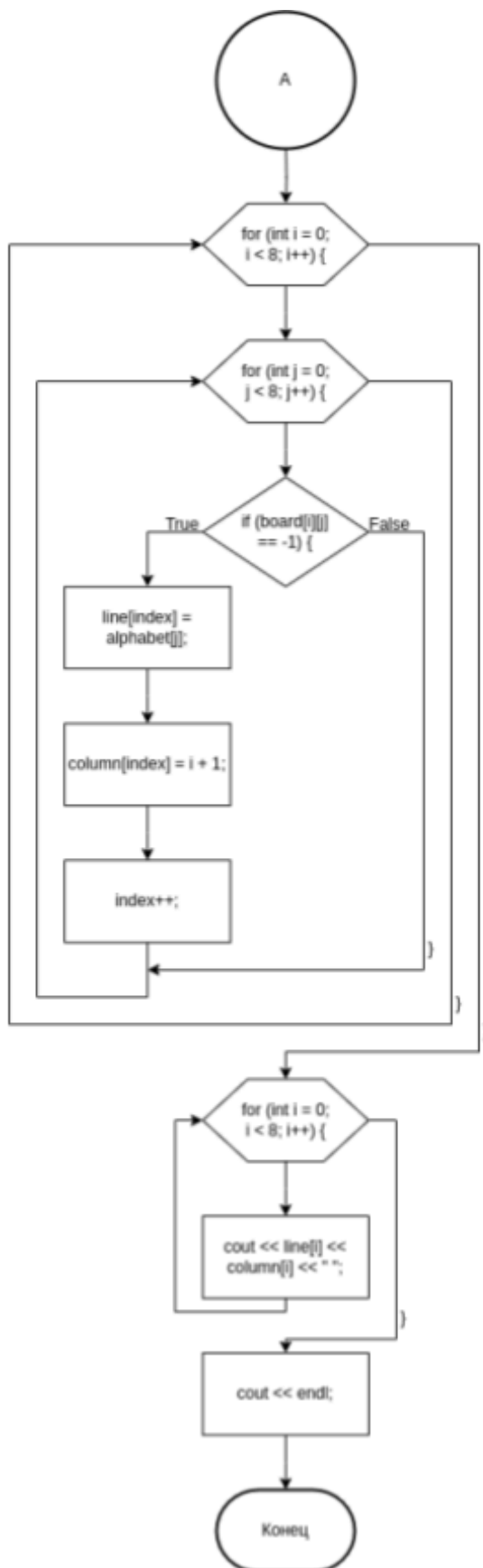


Рисунок 5



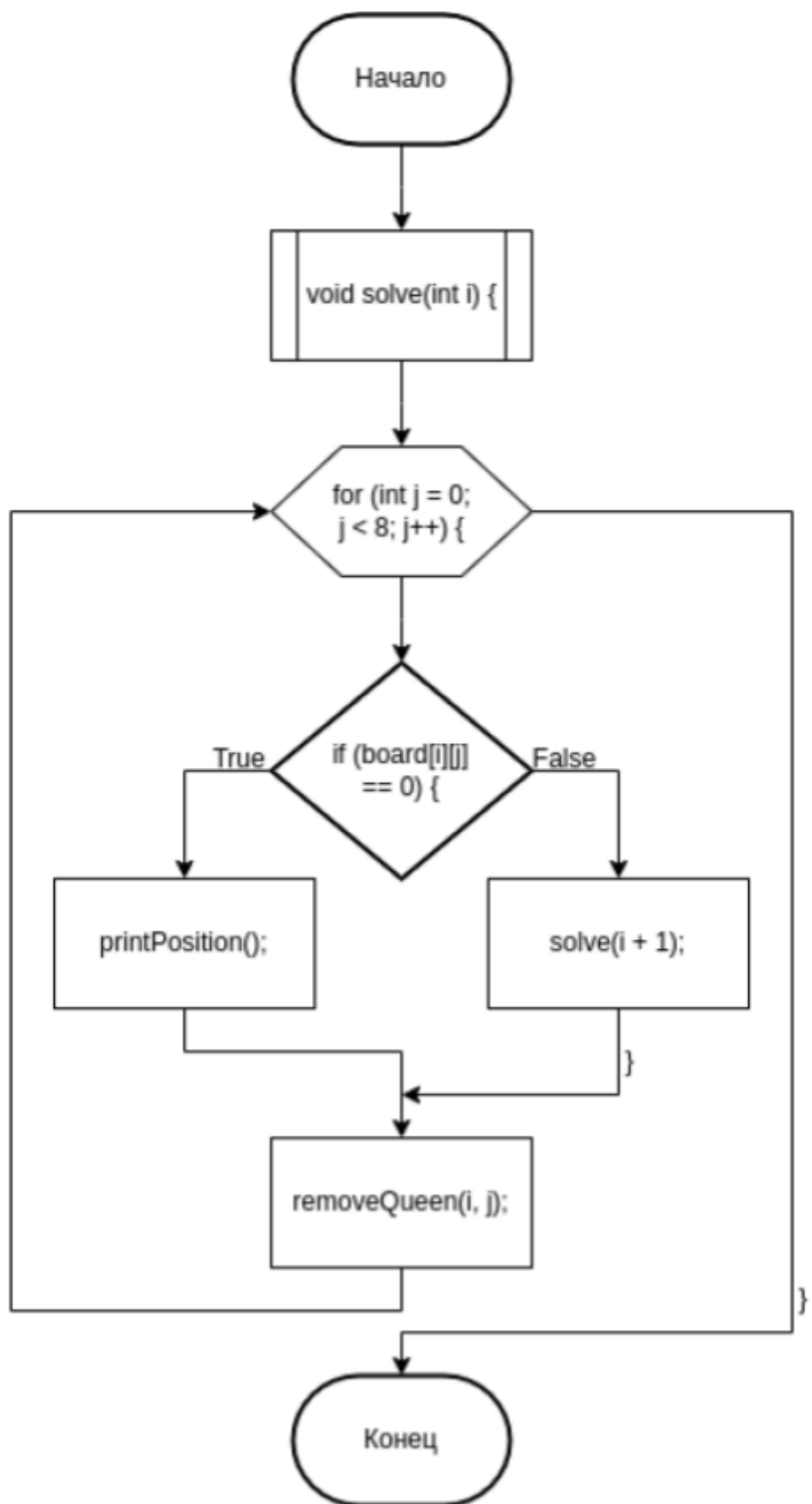


Рисунок 6

## 2 Код программы

```
#include <iostream>
using namespace std;

int board[8][8] = {0};

void setQueen(int i, int j);
void removeQueen(int i, int j);
void printPosition();
void solve(int i);

int main() {
    solve(0);
    return 0;
}

void setQueen(int i, int j) {
    for (int x = 0; x < 8; x++) {
        board[x][j]++;
        board[i][x]++;
        if (i + j - x >= 0 && i + j - x < 8) {
            board[i + j - x][x]++;
        }
        if (i - j + x >= 0 && i - j + x < 8) {
            board[i - j + x][x]++;
        }
    }
}
```

```

        board[i][j] = -1;
    }

```

```

void removeQueen(int i, int j) {
    for (int x = 0; x < 8; x++) {
        board[x][j]--;
        board[i][x]--;
        if (i + j - x >= 0 && i + j - x < 8) {
            board[i + j - x][x]--;
        }
        if (i - j + x >= 0 && i - j + x < 8) {
            board[i - j + x][x]--;
        }
    }
}

```

```

    board[i][j] = 0;
}

```

```

void printPosition() {
    char alphabet[8] = {'a', 'b', 'c', 'd', 'e',
    'f', 'g', 'h'};
    char line[8];
    int column[8];
    int index = 0;
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            if (board[i][j] == -1) {
                line[index] = alphabet[j];
                column[index] = i + 1;
                index++;
            }
        }
    }
}

```

```

        }
    }
}

for (int i = 0; i < 8; i++) {
    cout << line[i] << column[i] << " ";
}
cout << endl;
}

```

```

void solve(int i) {
    for (int j = 0; j < 8; j++) {
        if (board[i][j] == 0) {
            setQueen(i, j);
            if (i == 7) {
                printPosition();
            }
            else {
                solve(i + 1);
            }
            removeQueen(i, j);
        }
    }
}

```

### 3 Работа программы

Вывод:

```

a1 e2 h3 f4 c5 g6 b7 d8
a1 f2 h3 c4 g5 d6 b7 e8
a1 g2 d3 f4 h5 b6 e7 c8
a1 g2 e3 h4 b5 d6 f7 c8
b1 d2 f3 h4 c5 a6 g7 e8

```

b1 e2 g3 a4 c5 h6 f7 d8  
b1 e2 g3 d4 a5 h6 f7 c8  
b1 f2 a3 g4 d5 h6 c7 e8  
b1 f2 h3 c4 a5 d6 g7 e8  
b1 g2 c3 f4 h5 e6 a7 d8  
b1 g2 e3 h4 a5 d6 f7 c8  
b1 h2 f3 a4 c5 e6 g7 d8  
c1 a2 g3 e4 h5 b6 d7 f8  
c1 e2 b3 h4 a5 g6 d7 f8  
c1 e2 b3 h4 f5 d6 g7 a8  
c1 e2 g3 a4 d5 b6 h7 f8  
c1 e2 h3 d4 a5 g6 b7 f8  
c1 f2 b3 e4 h5 a6 g7 d8  
c1 f2 b3 g4 a5 d6 h7 e8  
c1 f2 b3 g4 e5 a6 h7 d8  
c1 f2 d3 a4 h5 e6 g7 b8  
c1 f2 d3 b4 h5 e6 g7 a8  
c1 f2 h3 a4 d5 g6 e7 b8  
c1 f2 h3 a4 e5 g6 b7 d8  
c1 f2 h3 b4 d5 a6 g7 e8  
c1 g2 b3 h4 e5 a6 d7 f8  
c1 g2 b3 h4 f5 d6 a7 e8  
c1 h2 d3 g4 a5 f6 b7 e8  
d1 a2 e3 h4 b5 g6 c7 f8  
d1 a2 e3 h4 f5 c6 g7 b8  
d1 b2 e3 h4 f5 a6 c7 g8  
d1 b2 g3 c4 f5 h6 a7 e8  
d1 b2 g3 c4 f5 h6 e7 a8  
d1 b2 g3 e4 a5 h6 f7 c8  
d1 b2 h3 e4 g5 a6 c7 f8  
d1 b2 h3 f4 a5 c6 e7 g8  
d1 f2 a3 e4 b5 h6 c7 g8

d1 f2 h3 b4 g5 a6 c7 e8  
d1 f2 h3 c4 a5 g6 e7 b8  
d1 g2 a3 h4 e5 b6 f7 c8  
d1 g2 c3 h4 b5 e6 a7 f8  
d1 g2 e3 b4 f5 a6 c7 h8  
d1 g2 e3 c4 a5 f6 h7 b8  
d1 h2 a3 c4 f5 b6 g7 e8  
d1 h2 a3 e4 g5 b6 f7 c8  
d1 h2 e3 c4 a5 g6 b7 f8  
e1 a2 d3 f4 h5 b6 g7 c8  
e1 a2 h3 d4 b5 g6 c7 f8  
e1 a2 h3 f4 c5 g6 b7 d8  
e1 b2 d3 f4 h5 c6 a7 g8  
e1 b2 d3 g4 c5 h6 f7 a8  
e1 b2 f3 a4 g5 d6 h7 c8  
e1 b2 h3 a4 d5 g6 c7 f8  
e1 c2 a3 f4 h5 b6 d7 g8  
e1 c2 a3 g4 b5 h6 f7 d8  
e1 c2 h3 d4 g5 a6 f7 b8  
e1 g2 a3 c4 h5 f6 d7 b8  
e1 g2 a3 d4 b5 h6 f7 c8  
e1 g2 b3 d4 h5 a6 c7 f8  
e1 g2 b3 f4 c5 a6 d7 h8  
e1 g2 b3 f4 c5 a6 h7 d8  
e1 g2 d3 a4 c5 h6 f7 b8  
e1 h2 d3 a4 c5 f6 b7 g8  
e1 h2 d3 a4 g5 b6 f7 c8  
f1 a2 e3 b4 h5 c6 g7 d8  
f1 b2 g3 a4 c5 e6 h7 d8  
f1 b2 g3 a4 d5 h6 e7 c8  
f1 c2 a3 g4 e5 h6 b7 d8  
f1 c2 a3 h4 d5 b6 g7 e8

```

f1 c2 a3 h4 e5 b6 d7 g8
f1 c2 e3 g4 a5 d6 b7 h8
f1 c2 e3 h4 a5 d6 b7 g8
f1 c2 g3 b4 d5 h6 a7 e8
f1 c2 g3 b4 h5 e6 a7 d8
f1 c2 g3 d4 a5 h6 b7 e8
f1 d2 a3 e4 h5 b6 g7 c8
f1 d2 b3 h4 e5 g6 a7 c8
f1 d2 g3 a4 c5 e6 b7 h8
f1 d2 g3 a4 h5 b6 e7 c8
f1 h2 b3 d4 a5 g6 e7 c8
g1 a2 c3 h4 f5 d6 b7 e8
g1 b2 d3 a4 h5 e6 c7 f8
g1 b2 f3 c4 a5 d6 h7 e8
g1 c2 a3 f4 h5 e6 b7 d8
g1 c2 h3 b4 e5 a6 f7 d8
g1 d2 b3 e4 h5 a6 c7 f8
g1 d2 b3 h4 f5 a6 c7 e8
g1 e2 c3 a4 f5 h6 b7 d8
h1 b2 d3 a4 g5 e6 c7 f8
h1 b2 e3 c4 a5 g6 d7 f8
h1 c2 a3 f4 b5 e6 g7 d8
h1 d2 a3 c4 f5 b6 g7 e8

```

#### 4 Объяснение функций

Функция `void setQueen(int i, int j)` - функция, устанавливающая ферзя на клетку с координатами `i, j`. Прибавляет единицу к клеткам доски, находящимся на одной вертикали, горизонтали и обеих диагоналях с ферзем. Ферзь обозначается «-1».

Функция `void removeQueen(int i, int j)` - функция, удаляющая ферзя на клетке с координатами `i, j`.

Функция `printPosition()` - функция, отображающая найденное решение. Каждое отдельное решение находится на отдельной строке и содержит 8 шахматных координат - координаты расположения ферзей.

Функция `void solve(i)` - основная функция, вызывающая себя рекурсивно. Эта функция находит решения для *i*-той строки.

### Структура репозитория на GitHub

Структура репозитория показана на рисунке 7.