Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе № 13**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Рекурсивные функции. Задача о расстановке восьми ферзей

Вариант 21

Выполнил:

Студент группы РИС-20-1б

Ознобихин Елисей Андреевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

1. Решить задачу о восьми ферзях с помощью алгоритма поиска методом возврата.

**Постановка задачи**

Дано шахматное поле размером 8x8. Необходимо поставить 8 ферзей таким образом, чтобы ни один ферзь не был под боем какого-либо другого ферзя.

**Анализ задачи**

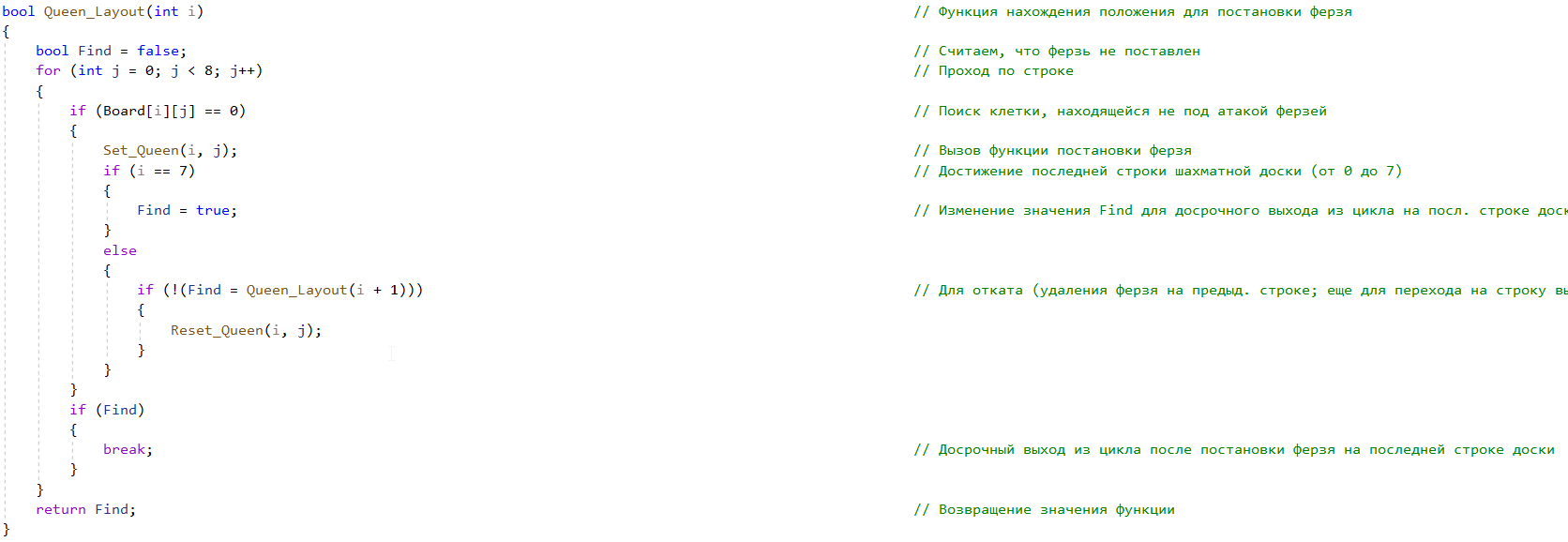
1. Для решения задачи необходимо:
   1. Реализовать функцию постановки ферзя на доску и вычисление клеток под боем.



* 1. Разработать функцию удаления ферзя с доски с учетом освободившихся клеток, которые были под боем.



* 1. Разработать функцию нахождения свободной клетки для постановки ферзя.



* 1. Реализовать применение этих функций в главной функции.



1. В ходе работы были использованы типы данных:
   1. Для функции Set\_Queen используются следующие аргументы:
   2. Тип int: индекс строки двумерного множества.
   3. Тип int: индекс столбца двумерного множества.

Сама функция имеет тип указателя на People и возвращает заполненное множество.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_Z6A0QvGKC7.png

* 1. Для функции Reset\_Queen используются следующие аргументы:
  2. Тип int: индекс строки двумерного множества.
  3. Тип int: индекс столбца двумерного множества.

Сама функция имеет тип void, поскольку при работе функции не нужно возвращать значение.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_aQUkYbbDOO.png

* 1. Для функции Queen\_Layout используются следующие аргументы:
  2. Тип int: индекс строки двумерного множества.

Сама функция имеет тип bool и возвращает истину, если параметр при полном выполнении функции равен 7.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_q3REcZAfvq.png

1. Для решения задачи данные были представлены в следующем виде:
   1. Для работы данные были записаны в целочисленном формате в двумерное множество Board, имитирующую шахматную доску. Переменная Board является глобальной переменной.

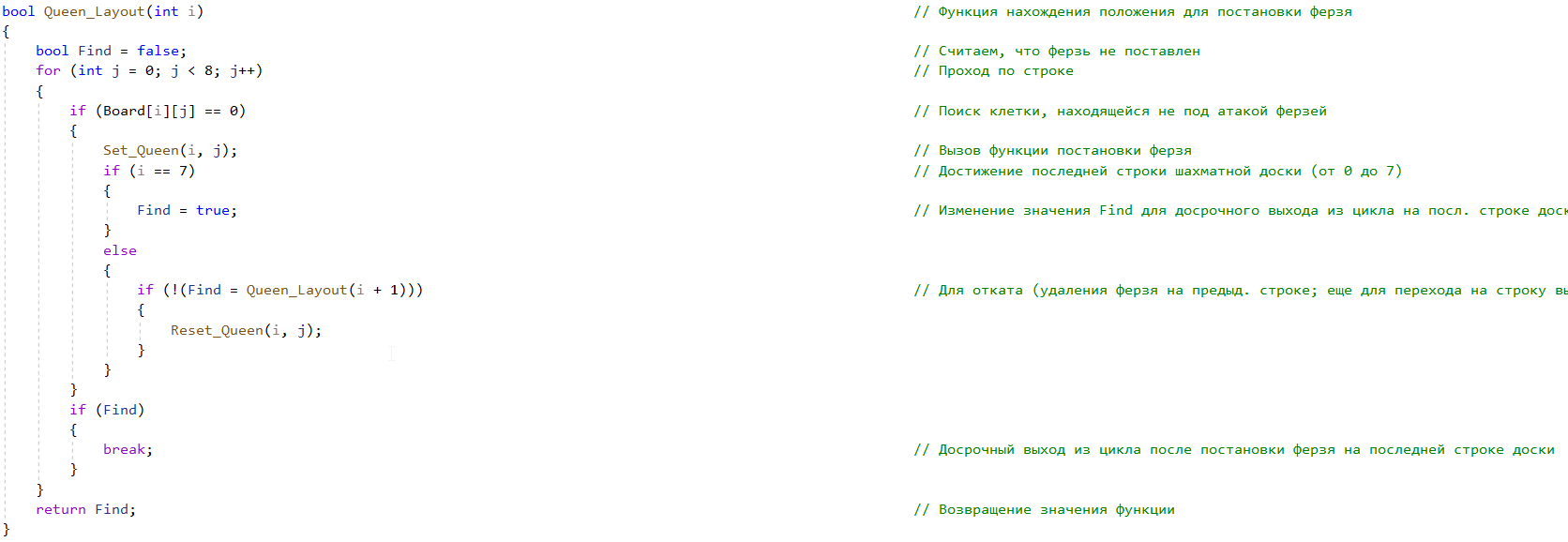
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_b7nISqBpo7.png

1. Для операций ввода и вывода использовались следующие операторы и функции:
   1. Операции ввода не требуется, поскольку пользователю не требуется входных данных.
   2. Вывод данных на консоль реализован с помощью оператора cout. Символ “+” имитирует клетку, на которую поставлен ферзь, символ “.” имитирует пустую клетку.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_SfxMyUfyXr.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_6Owv9aVNEC.png

1. Поставленные задачи решены следующими действиями:
   1. Для реализации Метода Поиска с Возвратом использовались вручную написанные функции Queen\_Layout, Reset\_Queen и Set\_Queen, выполняющие поиск свободной клетки для постановки ферзя, удаления ферзя с клетки доски и постановки ферзя на клетку доски.







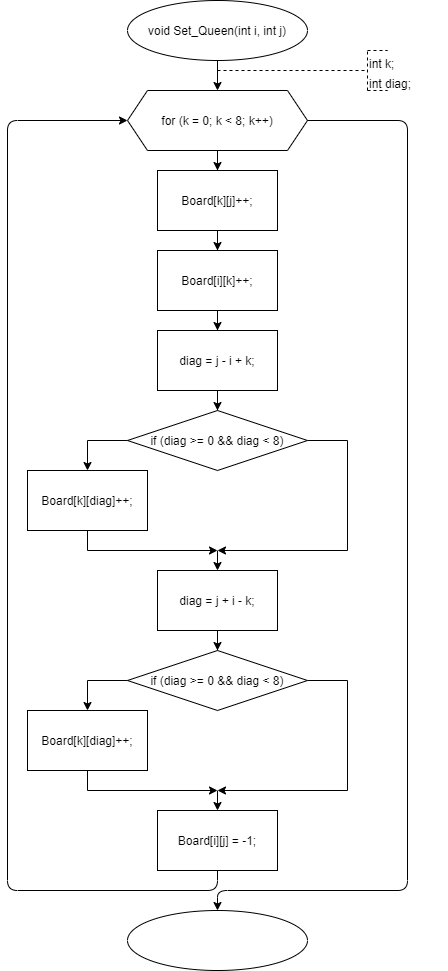
* 1. При работе функции Reset\_Queen выполняется цикл, в котором при каждой итерации уменьшает значения четырёх элементов на 1. Циклом проходятся вертикаль, горизонталь и две диагонали двумерного множества.

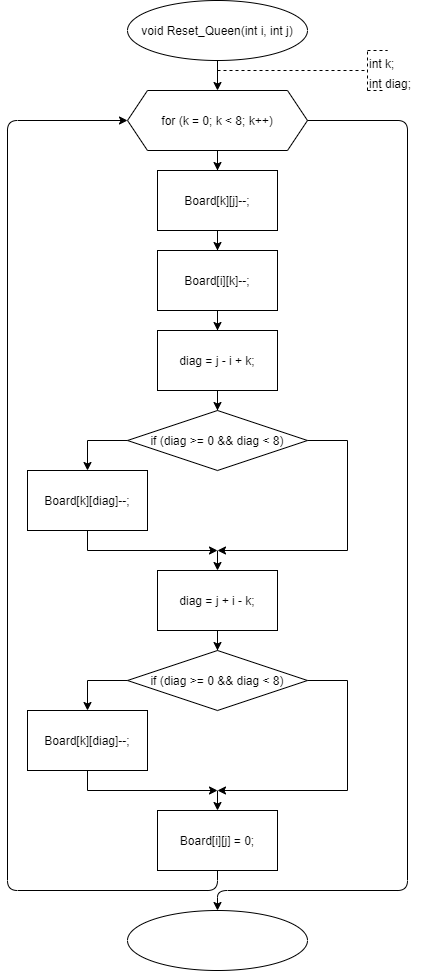


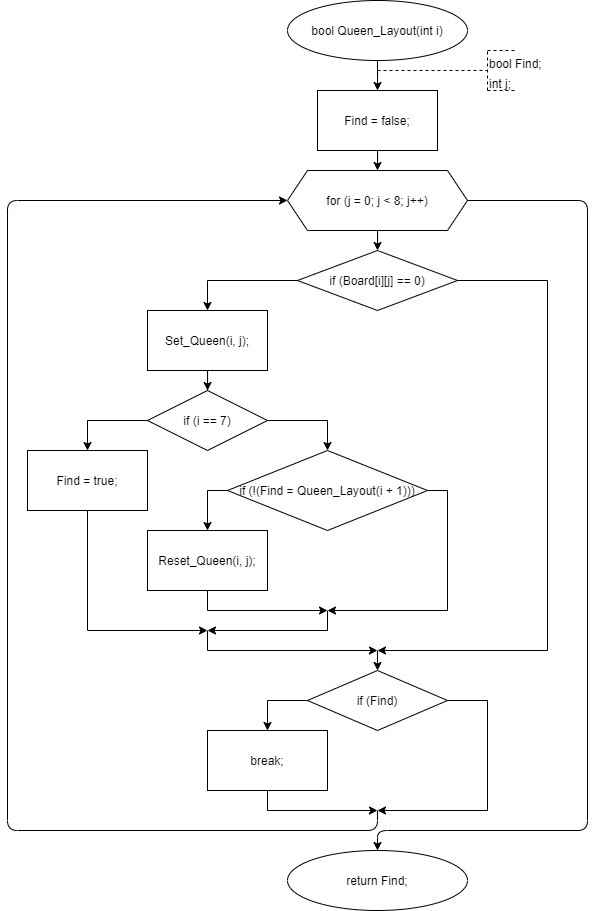
* 1. При работе функции Set\_Queen выполняется цикл, в котором при каждой итерации увеличивает значения четырёх элементов на 1. Циклом проходятся вертикаль, горизонталь и две диагонали двумерного множества.

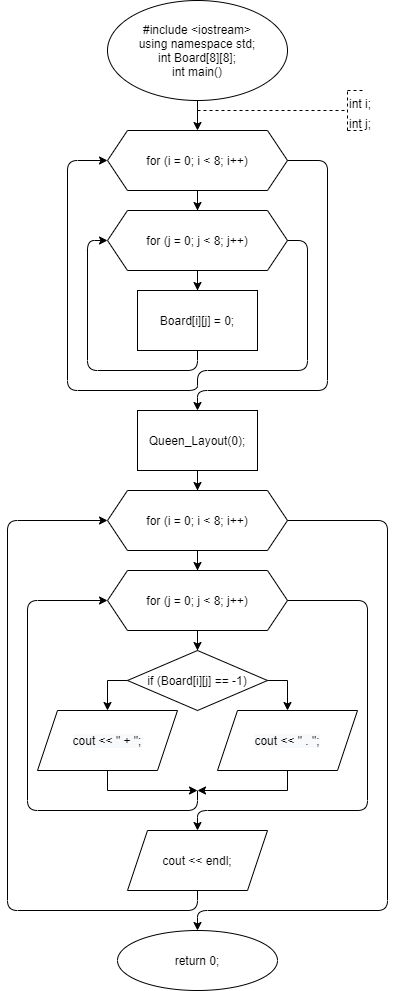


**Блок-схема программы**









**Реализация задачи на языке С++**

#include <iostream>

using namespace std;

int Board[8][8];

void Set\_Queen(int i, int j)

{

for (int k = 0; k < 8; k++)

{

Board[k][j]++;

Board[i][k]++;

int diag;

diag = j - i + k;

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

{

Board[k][diag]++;

}

diag = j + i - k;

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

{

Board[k][diag]++;

}

Board[i][j] = -1;

}

}

void Reset\_Queen(int i, int j)

{

for (int k = 0; k < 8; k++)

{

Board[k][j]--;

Board[i][k]--;

int diag;

diag = j - i + k;

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

{

Board[k][diag]--;

}

diag = j + i - k;

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

{

Board[k][diag]--;

}

Board[i][j] = 0;

}

}

bool Queen\_Layout(int i)

{

bool Find = false;

for (int j = 0; j < 8; j++)

{

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

{

Set\_Queen(i, j);

if (i == 7)

{

Find = true;

}

else

{

if (!(Find = Queen\_Layout(i + 1)))

Reset\_Queen(i, j);

}

}

}

if (Find)

{

break;

}

}

return Find;

}

int main()

{

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++)

{

Board[i][j] = 0;

}

}

Queen\_Layout(0);

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++)

{

if (Board[i][j] == -1)

{

cout << " + ";

}

else

{

cout << " . ";

}

}

cout << endl;

}

}

**Скриншоты результатов работы программы**

