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 подсчётом

Вариант 17

Выполнил:

Студент группы Рис-20-1б

Томилов Владислав Алексеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

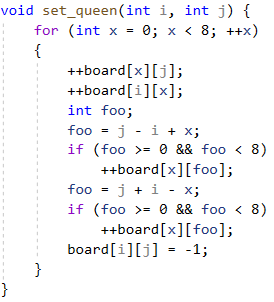
**Постановка задачи**

Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого

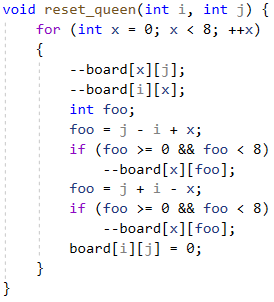
**Анализ задачи**

**1.** Для решения задачи необходимо:

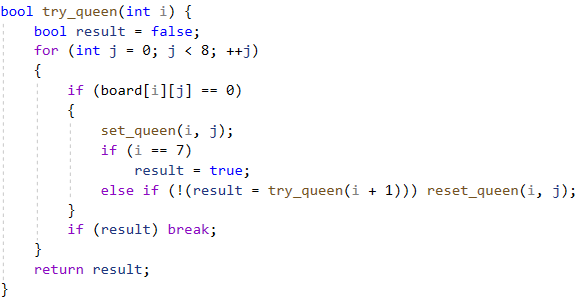
**1.1.** Разработать функцию set\_queen для постановки ферзя. После проверки условия в клетку ставится ферзь, затем прибавляется 1 к значениям клеток по горизонтали и вертикали от данной клетки, а также все ко всем клеткам по диагонали.



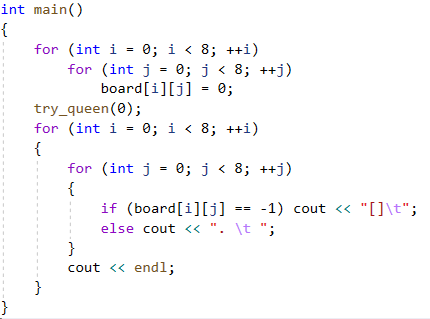
**1.2.** Разработать функцию reset\_queen для удаления ферзя. После проверки условия из клетки удаляется ферзь, затем убавляется 1 у значений клеток по горизонтали и вертикали от данной клетки, а также у всех клеток по диагонали.



**1.3.** Разработать функцию try\_queen, которая проверяет, можно ли поставить ферзя на данную клетку. Если значение клетки 0, это значит, что она пуста и не находится под боем, следовательно ферзя поставить можно. Если же в строке нет таких клеток, программ переходит на предыдущую строку удаляет в ней ферзя и ищет новое место для ферзя в предыдущей строке.



**1.4.** Основная функция int main(). В этой функции сначала происходит обнуление всех ячеек шахматной доски, затем запускается рекурсивная функция try\_queen, и в итоге выводятся значения всех клеток доски.



**2.**В ходе работы были использованы следующие типы данных:

**2.1.** Все ячейки шахматной доски имеют тип данных int.

int board[8][8];

**2.2.** Для функции set\_queen используются следующие аргументы:

1. Тип int: Номер строки, в которой следует установить ферзя.

2. Тип int: Номер столбца, в которой следует установить ферзя.



Сама функция имеет тип void, так как в ходе ее работы нет необходимости возвращать значение для этой функции.

**2.3.** Для функции reset\_queen пользуются следующие аргументы:

1. Тип int: Номер строки, в которой следует удалить ферзя.

2. Тип int: Номер столбца, в которой следует удалить ферзя.



Сама функция имеет тип void, так как в ходе ее работы нет необходимости возвращать значение для этой функции.

**2.4.** Для функции try\_queen используются следующие аргументы:

1. Тип int: Номер строки, с которой начинается проверка на постановку ферзя.



Сама функция имеет тип bool, так как в ходе ее работы возвращается значение переменной result, которая является условием выхода из рекурсии, если мы достигли последней клетки доски и поставили в ней ферзя, следовательно все ферзи расставлены и надо выходить из цикла.

**Код программы**

#include <iostream>

using namespace std;

int board[8][8];

void set\_queen(int i, int j) {

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

{

++board[x][j];

++board[i][x];

int foo;

foo = j - i + x;

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

++board[x][foo];

foo = j + i - x;

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

++board[x][foo];

board[i][j] = -1;

}

}

void reset\_queen(int i, int j) {

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

{

--board[x][j];

--board[i][x];

int foo;

foo = j - i + x;

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

--board[x][foo];

foo = j + i - x;

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

--board[x][foo];

board[i][j] = 0;

}

}

bool try\_queen(int i) {

bool result = false;

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

{

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

{

set\_queen(i, j);

if (i == 7)

result = true;

else if (!(result = try\_queen(i + 1))) reset\_queen(i, j);

}

if (result) break;

}

return result;

}

int main()

{

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

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

board[i][j] = 0;

try\_queen(0);

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

{

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

{

if (board[i][j] == -1) cout << "[]\t";

else cout << ". \t ";

}

cout << endl;

}

}