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: «Информатика»

Тема: Рекурсивные функции. Задача о расстановке восьми ферзей

Вариант 13

Выполнил:

Студент группы РИС-20-1б

Кузнецов Михаил Сергеевиич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Цель работы**

1. Алгоритмическим способом решить задачу про расстановку восьми ферзей на шахматной доске.
2. Программа должна вывести правильную расстановку восьми ферзей на доске так, чтобы ни один ферзь не находился под ударом другого.

**Постановка задачи**

1) Необходимо разработать программу, которая создаёт шахматную доску в виде двумерного массива и выполняет постановку 8 ферзей с учётом правил игры в шахматы.

2) В результате работы в консоль будут выведены все шаги программы.

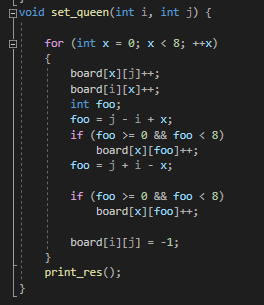
**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи.

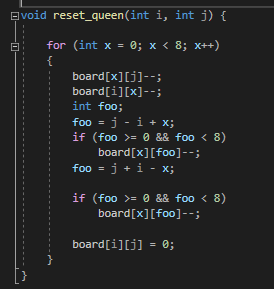
* Создание игрового поля.
* Создание рекурсивной функции для постановки ферзей.
* В случае не верного хода, программа должна откатывать последние шаги.
* Вывести все шаги в консоль.

2) Игровое поле:

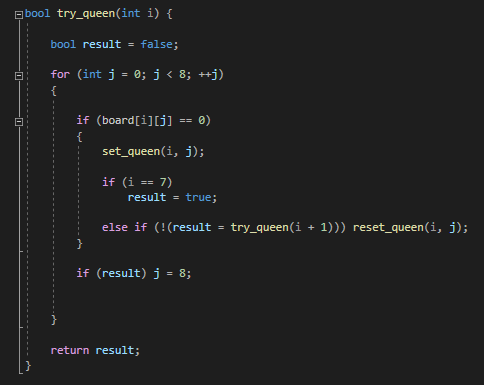
  
3) Постановка ферзя.



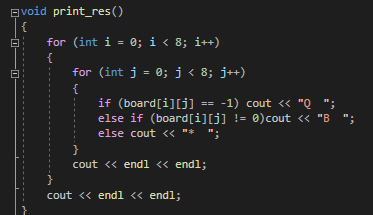
4) Удаление ферзя:



5) Основная рекурсивная функция:



6) Вывод результата:



**Код**

#include <iostream>

using namespace std;

int board[8][8];

void print\_res()

{

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

{

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

{

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

else if (board[i][j] != 0)cout << "B ";

else cout << "\* ";

}

cout << endl << endl;

}

cout << endl << endl;

}

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;

}

print\_res();

}

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) j = 8;

}

return result;

}

int main()

{

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

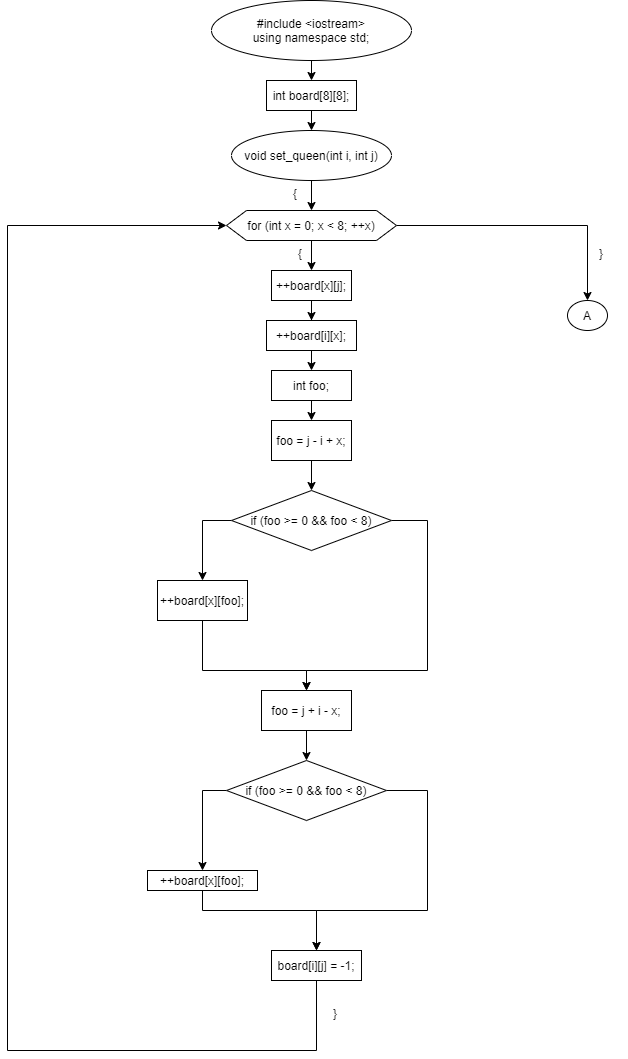
for (int j = 0; j < 8; ++j)

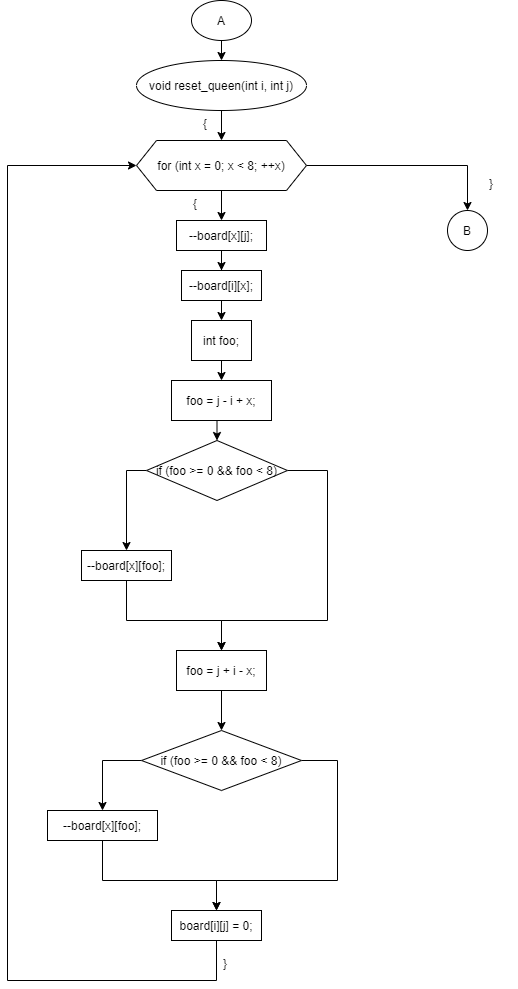
board[i][j] = 0;

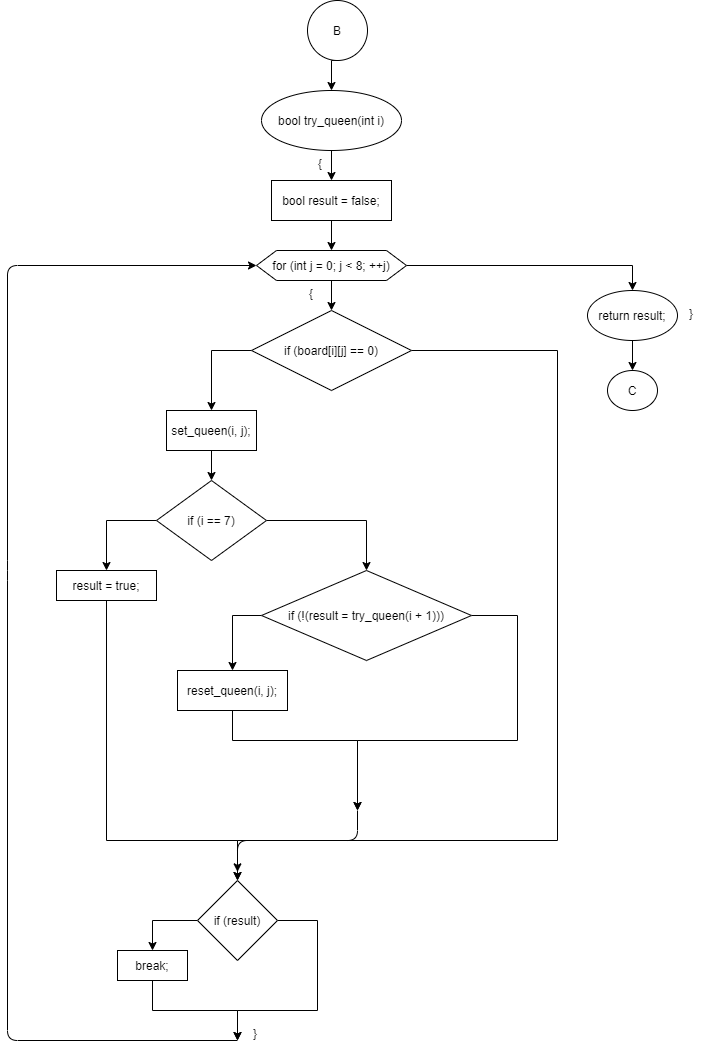
try\_queen(0);

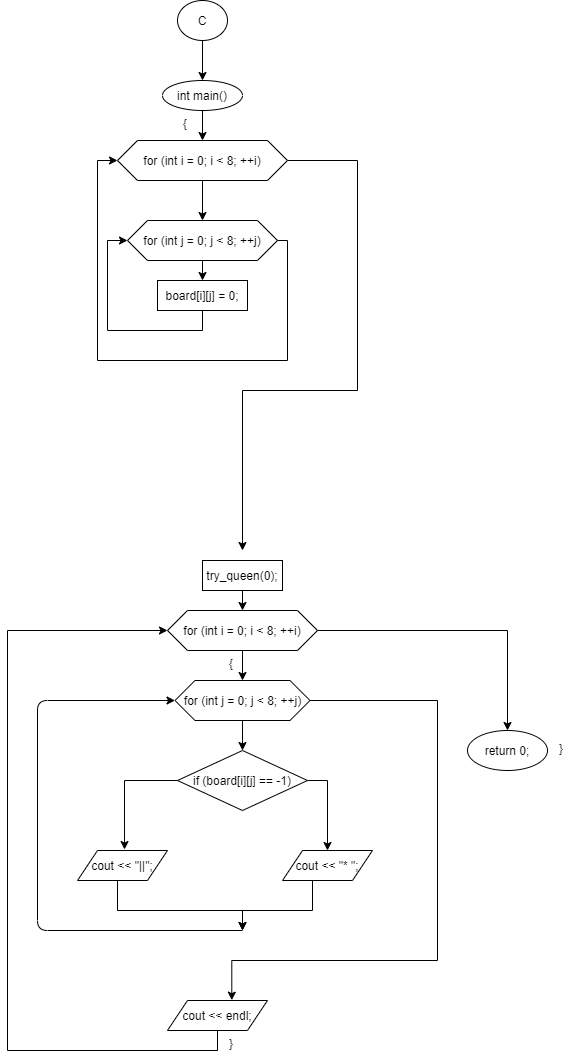
}

**Блок схема**

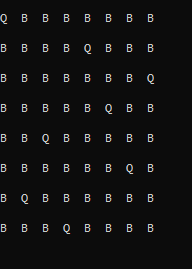








**Скриншоты тестов**

****