**«Пермский национальный исследовательский  
политехнический университет»**

**О Т Ч Ё Т**

**по лабораторной работе №13**

Тема: Рекурсивные функции. Задача о расстановке восьми ферзей

Выполнил работу

студент группы ИВТ-20-2б

Копытов А.А.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

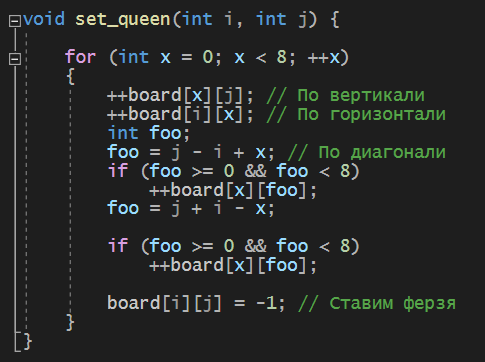
Пермь, 2021

**Постановка задачи**

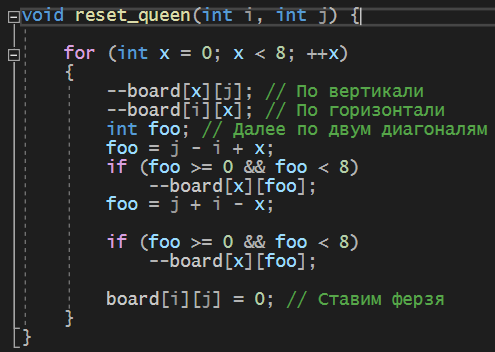
1. Алгоритмическим способом решить задачу про расстановку восьми ферзей на шахматной доске.
2. Программа должна вывести правильную расстановку восьми ферзей на доске так, чтобы ни один ферзь не находился под ударом другого.

**Анализ задачи**

1. Для решения задачи необходимо…
   1. Создать функцию set\_queen типа void, которая занимается расстановкой ферзей в консоли.



* 1. Реализовать функцию reset\_queen, которая проверяет правильность расстановки ферзей и в случае неверной расстановки возвращается обратно и редактирует предыдущие строки, чтобы в целом получить верную расстановку;

****

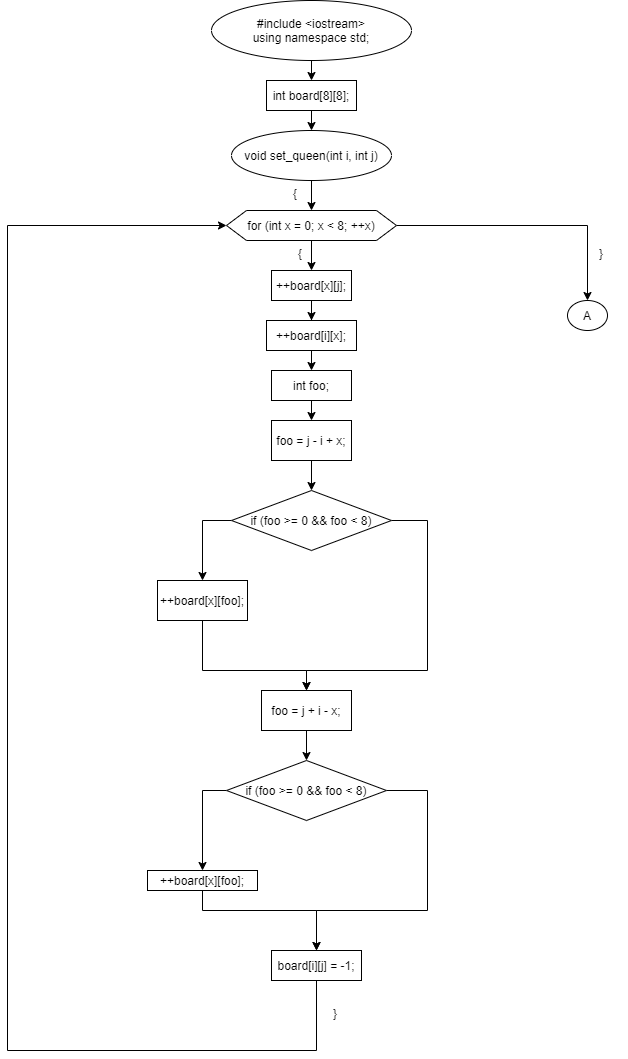
1. В программе были использованы следующие типы данных:
   1. Тип данных int, используемый в качестве счетчика цикла;

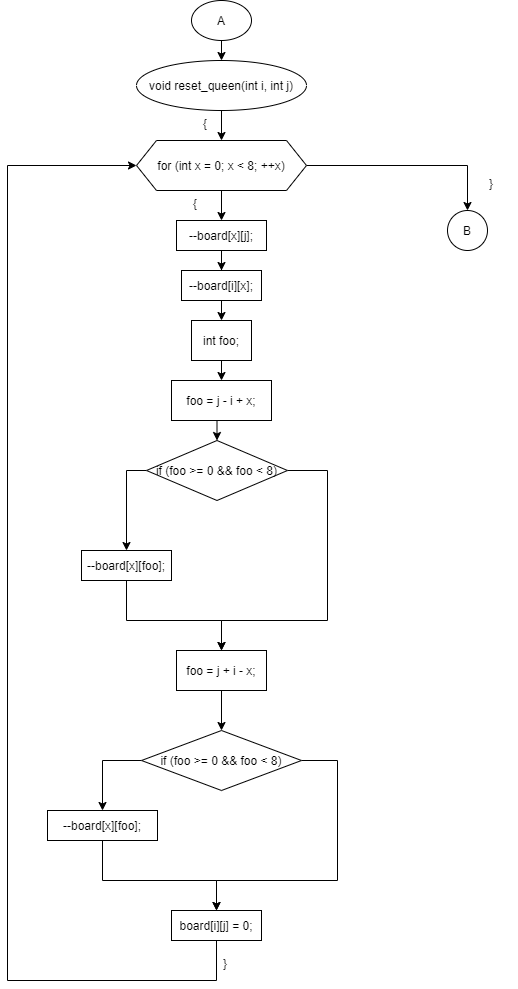


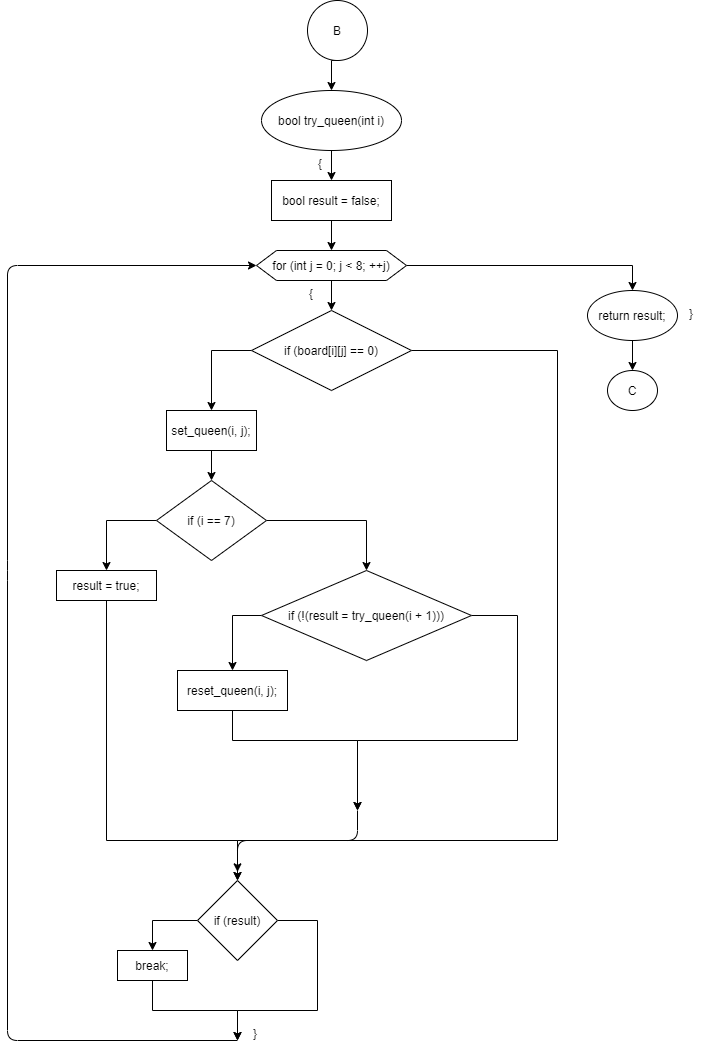
* 1. Тип данных bool для логического флажка, проверяющего правильность расстановки ферзей;

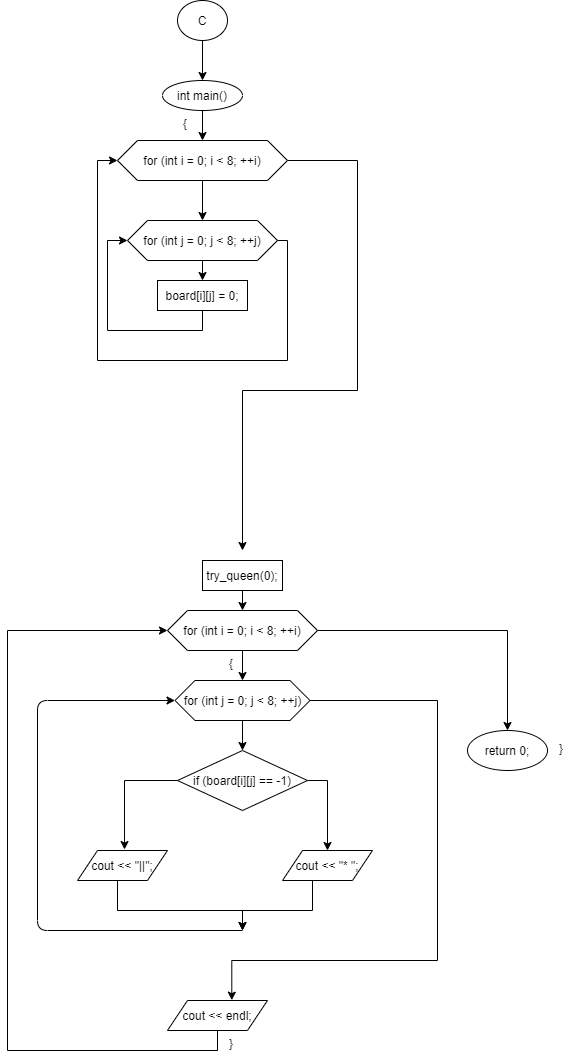


**Блок—схема программы**









**Решение**

#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]; // По вертикали

++board[i][x]; // По горизонтали

int foo;

foo = j - i + x; // По диагонали

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

++board[x][foo];

foo = j + i - x;

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

++board[x][foo];

board[i][j] = -1; // Ставим ферзя

}

}

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

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

{

--board[x][j]; // По вертикали

--board[i][x]; // По горизонтали

int foo; // Далее по двум диагоналям

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; // Ставим ферзя

}

}

bool try\_queen(int i) { // Получаем номер строки

bool result = false; // начальное значение

for (int j = 0; j < 8; ++j) // итерирование по КЛЕТКАМ строки

{

if (board[i][j] == 0) // поиск клетки, находящейся не под атакой ферзей

{

set\_queen(i, j); // вызов функции постановки ферзя И обозначения клеток, которые он бьет

if (i == 7) // достижение последней строки шахматной доски (от 0 до 7)

result = true; // изменение значения result для досрочного выхода из цикла на посл. строке доски после постановки ферзя (63 стр. кода)

else if (!(result = try\_queen(i + 1))) reset\_queen(i, j); /\* для отката(удаления ферзя на предыд. строке

еще для перехода на строку выше \*/

}

if (result) break; // досрочный выход из цикла после постановки ферзя на посл. строке доски

}

return result; // возвращение значения функции

}

int main()

{

for (int i = 0; i < 8; ++i) // Сначала обнуляем в

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

board[i][j] = 0;

try\_queen(0); // Запускаем цикл

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

{ // Выводим все содержимое

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

{

if (board[i][j] == -1) cout << "||"; // Так выводится ферзь

else cout << "\* "; // А так пустые клетки

}

cout << endl;

}

}

**Скриншоты**

