Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №13**

Дисциплина: основы алгоритмизации и программирования

Тема: “Метод поиска с возвратом” в C++. Ферзи

Выполнил работу

студент группы РИС-20-1б

Коваленко Н. А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Цель – написать программу для осуществления поиска с возвратом. Решение задачи о расстановки ферзей на шахматной доске.

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать алгоритмы прямого поиска с возвратом
* Реализация шахматной доски
* Написание функций для проверки постановки ферзя, постановки ферзя и удаления ферзя
* Написание главной функции для вызова других функций и вывода доски

Постановка задачи

Расставить на шахматной доске 8 ферзей

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание шахматной доски
* Создание функций для реализации действий с ферзями

void resetQ(int i, int j)

void setQ(int i, int j)

bool tryQ(int i)

* Реализация вывода доски и постановки ферзей

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

{

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

{

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

cout << "[]";

else

cout << ". ";

}

cout << endl;

}

1. Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами

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

Двоичный массив шахматной доски

int board[8][8];

1. Вывод данных осуществляется посредством функции

cout << ". ";

1. Для решения задачи будут использованы циклы. Например, цикл вывода доски

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

{

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

{

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

cout << "[]";

else

cout << ". ";

}

cout << endl;

}

Код

#include <iostream>

using namespace std;

int board[8][8];

void resetQ(int i, int j)

{

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

{

--board[x][j];

--board[i][x];

int k;

k = j - i + x;

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

--board[x][k];

k = j + i - x;

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

--board[x][k];

}

board[i][j] = 0;

}

void setQ(int i, int j)

{

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

{

++board[x][j];

++board[i][x];

int k;

k = j - i + x;

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

++board[x][k];

k = j + i - x;

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

++board[x][k];

}

board[i][j] = -1;

}

bool tryQ(int i)

{

bool result = false;

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

{

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

{

setQ(i, j);

if (i == 7)

result = true;

else

{

if (!(result = tryQ(i + 1)))

resetQ(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;

tryQ(0);

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

{

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

{

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

cout << "[]";

else

cout << ". ";

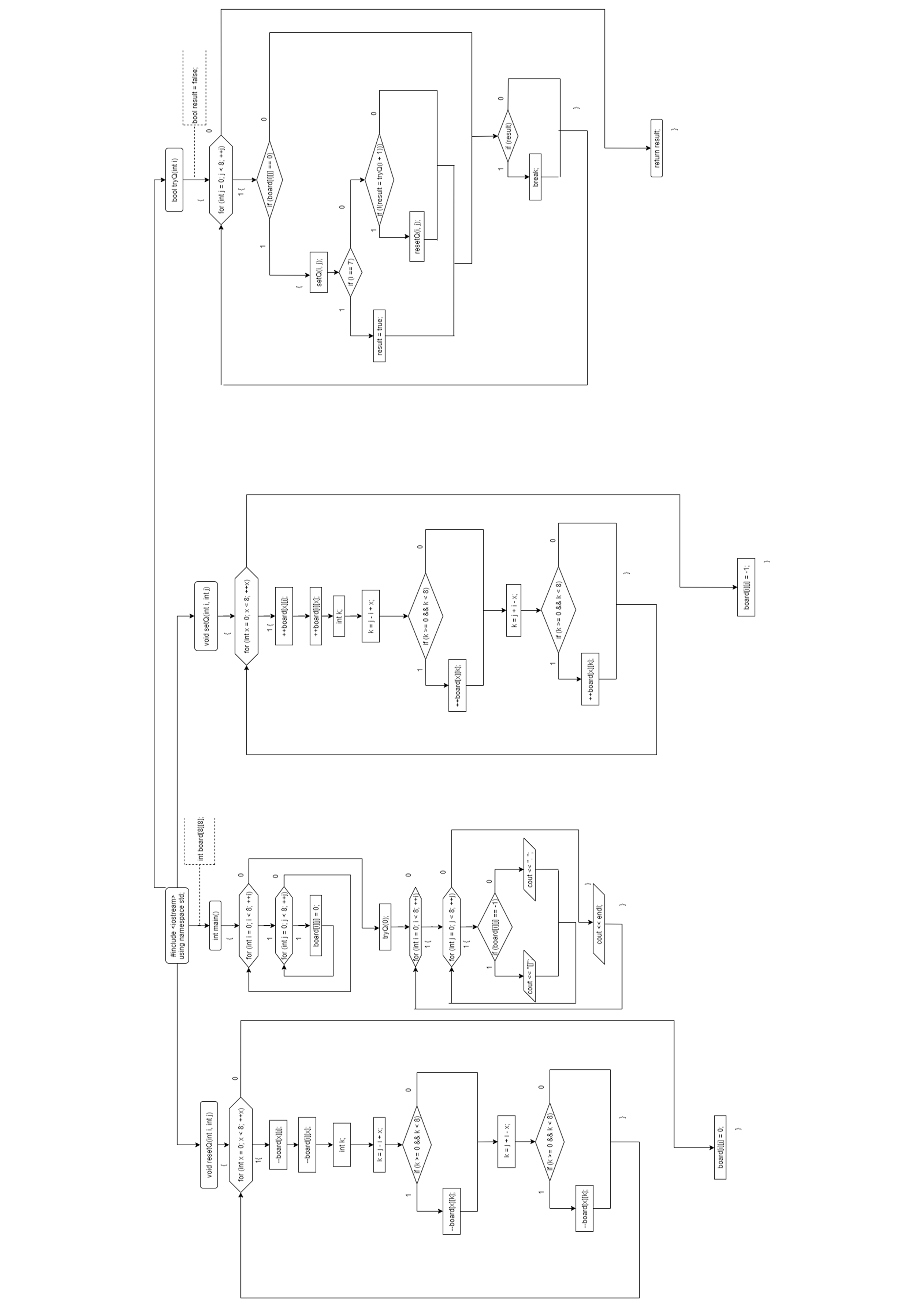
}

cout << endl;

}

}

Блок-схема:



Работа кода

