Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №13**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Рекурсивные функции. Задача о расстановке восьми ферзей

Вариант 18

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

# **Цель работы**

1) Получить практические навыки работы с функциями с возвратом;

# **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. Определить какие необходимо выполнить действия для решения задачи:

* Разработать игровое поле.

int board[8][8];

* Разработать функцию для постановки ферзей.

void SetQueen(int i, int j) {

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

++board[x][j]; //вертикаль

++board[i][x]; //горизонталь

int y;

//по диагоналям

y = j - i + x;

if ((y >= 0) && (y < 8)) {

++board[x][y];

}

y = j + i - x;

if ((y >= 0) && (y < 8)) {

++board[x][y];

}

board[i][j] = -1; //ставим ферзя

}

}

* Предусмотреть неверную расстановку, и откат последнего шага при помощи основной рекурсивной функции и функции удаления ферзя

bool TryQueen(int i) {

PrintBoard(board);

cout << endl;

bool result = false;

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

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

SetQueen(i, j);

if (i == 7)

result = true;

else if (!(result = TryQueen(i + 1))) {

RemoveQueen(i, j);

}

}

}

return result;

}

void RemoveQueen(int i, int j) {

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

--board[x][j]; // вертикаль

--board[i][x]; // горизонталь

int y;

//по диагоналям

y = j - i + x;

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

--board[x][y];

y = j + i - x;

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

--board[x][y];

board[i][j] = 0; // убираем ферзя

}

}

* Реализовать вывод каждого шага работы программы и финальную расстановку.

void PrintBoard(int board[8][8]) {

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 << "[.]";

}

else {

cout << "[ ]";

}

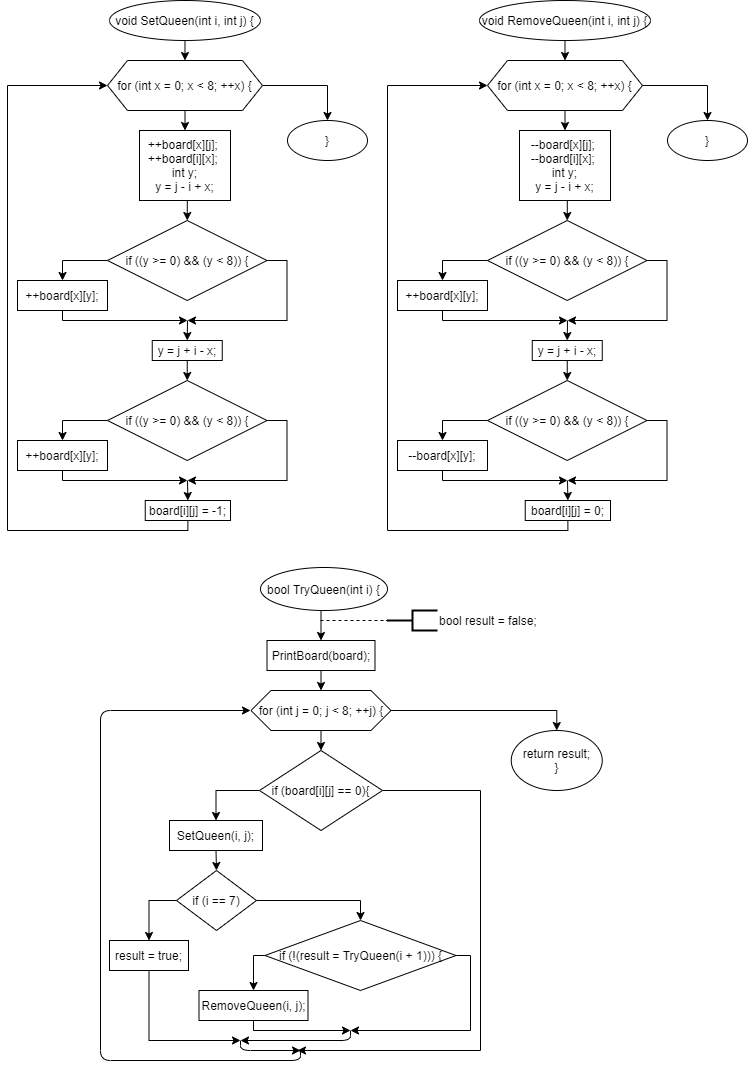
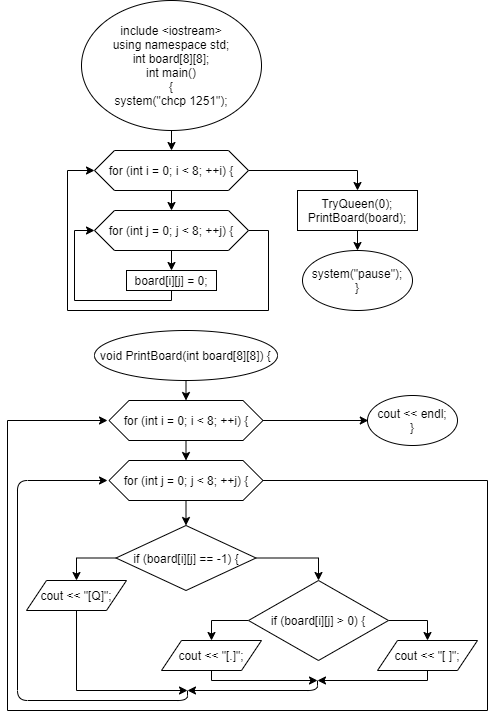
}

cout << endl;

}

}

# **Блок схема**



# **Код программы на языке C++**

#include <iostream>

using namespace std;

int board[8][8];

void PrintBoard(int board[8][8]) {

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 << "[.]";

}

else {

cout << "[ ]";

}

}

cout << endl;

}

}

void SetQueen(int i, int j) {

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

++board[x][j]; //вертикаль

++board[i][x]; //горизонталь

int y;

//по диагоналям

y = j - i + x;

if ((y >= 0) && (y < 8)) {

++board[x][y];

}

y = j + i - x;

if ((y >= 0) && (y < 8)) {

++board[x][y];

}

board[i][j] = -1; //ставим ферзя

}

}

void RemoveQueen(int i, int j) {

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

--board[x][j]; // вертикаль

--board[i][x]; // горизонталь

int y;

//по диагоналям

y = j - i + x;

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

--board[x][y];

y = j + i - x;

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

--board[x][y];

board[i][j] = 0; // убираем ферзя

}

}

bool TryQueen(int i) {

PrintBoard(board);

cout << endl;

bool result = false;

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

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

SetQueen(i, j);

if (i == 7)

result = true;

else if (!(result = TryQueen(i + 1))) {

RemoveQueen(i, j);

}

}

}

return result;

}

int main()

{

system("chcp 1251");

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

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

board[i][j] = 0;

}

}

cout << endl;

TryQueen(0);

PrintBoard(board);

cout << endl;

system("pause");

}

# **Скриншоты тестов**

