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.

Задача о восьми ферзях”

Выполнил:

студент группы ИВТ-20-2Б Тедеев А.З.

Проверила: доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Постановка задачи**

Используя метод поиска с возвратом, расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого.

2

**Анализ задачи**

**1.** Для решения задачи необходимо:

**1.1.** Организовать функцию set\_ferz(), которая будет ставить ферзя и менять клетки под боем.

**1.2.** Организовать функцию delete\_ferz(), которая будет удалять ферзя и очищать клетки под его боем.

**1.3.** Организовать функцию try\_ferz(), в которой методом возврата будут вызываться функции set\_ferz() и delete\_ferz() для решения поставленной задачи.

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

**2.1.** Двумерный статический массив doska типа int.

**2.2.** Переменная dia типа int для смещения по диагонали

**2.3.** Переменная result в качестве флажка.

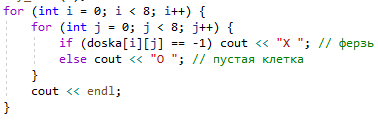
**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Через функцию try\_ferz отправляется позиция первого ферзя (первая клетка) в качестве параметра.

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** В качестве ввода используется число как параметр в функцию try\_ferz.

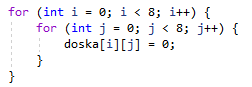
**4.2.** Для вывода используется цикл for в котором через функцию cout выводятся ферзи и пустые клетки.



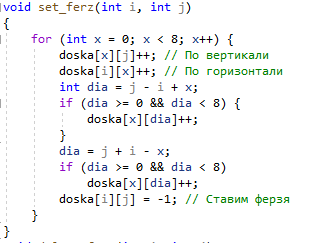
**5.** Поставленные задачи будут решены следующими действиями:

3

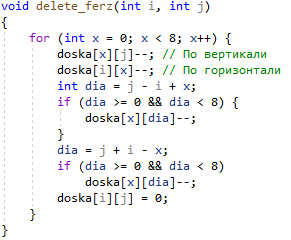
**5.1.** В главной функции все клетки доски заполняются нулями, что означает, что они пустые.



**5.2.** В функции try\_ferz() в цикле for происходит итерирование по строке на доске (по горизонтали), в ветвлении идёт проверка на то, является ли клетка под боем одного из ферзя, если она пустая, то вызывается функция set\_ferz(), в которой ставится ферзь (ферзь в памяти компьютера представляется как -1) и клетки под его боем (обозначаются инкрементом).

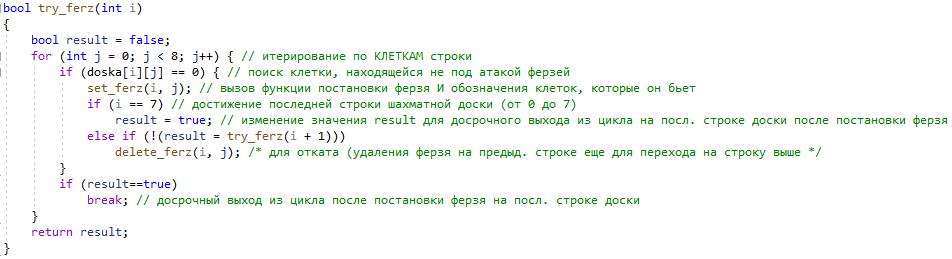


После обработки функции set\_ferz() используется ветвление, в котором проверяется, на каком по счёту ряду происходит итерация: если на последнем (7), то функция try\_ferz() может быть досрочно завершена через переменную result, иначе происходит вызов функции try\_ferz() с переходом на другой ряд рекурсивно, пока не будет достигнут последний ряд. Если на каком-то из рядов не получается поставить ферзя, то вызывается функция delete\_ferz().



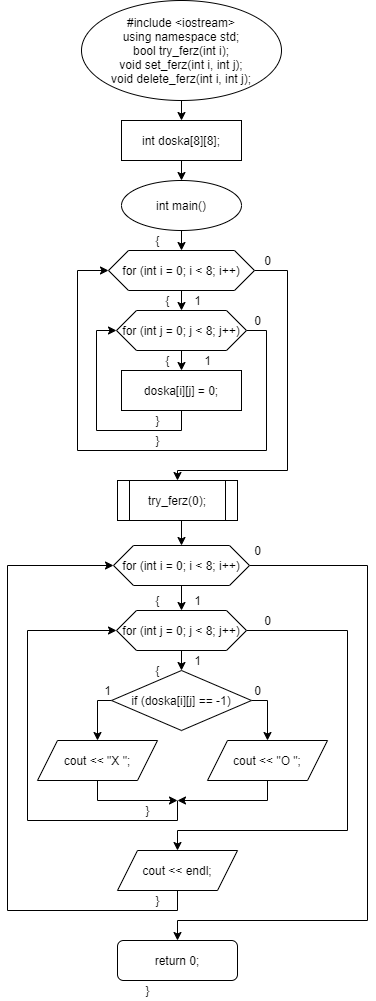
4

В функции delete\_ferz() происходит декремент клеток под боем ферзя, которого необходимо удалить, а также удаление самого ферзя.

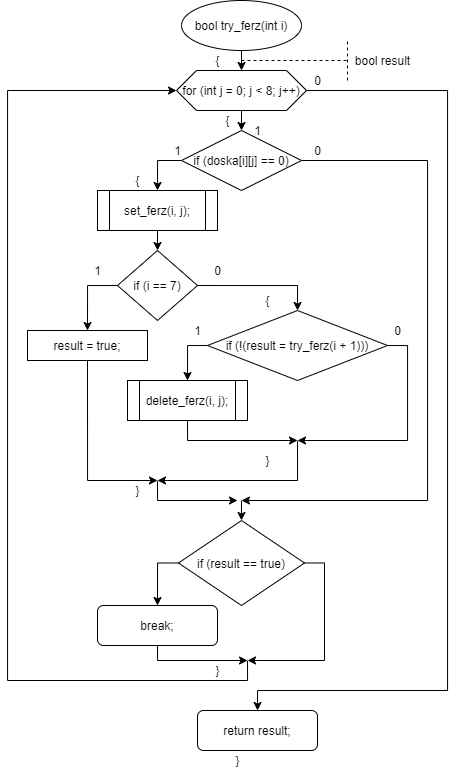


5

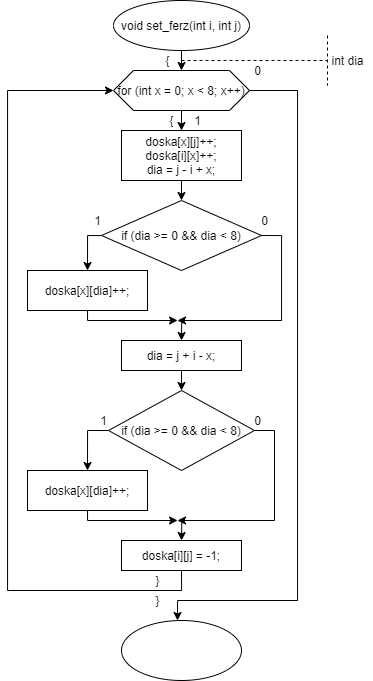
**Блок-схема**

****

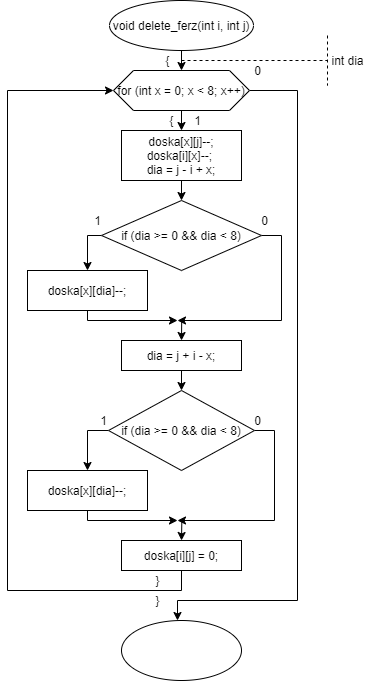
6

****

7

****

8

****

9

**Код**

#include <iostream>

using namespace std;

int doska[8][8];

bool try\_ferz(int i);

void set\_ferz(int i, int j);

void delete\_ferz(int i, int j);

int main()

{

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

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

doska[i][j] = 0;

}

}

try\_ferz(0);

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

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

if (doska[i][j] == -1) cout << "X "; // ферзь

else cout << "O "; // пустая клетка

}

cout << endl;

}

}

bool try\_ferz(int i)

{

bool result = false;

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

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

set\_ferz(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

else if (!(result = try\_ferz(i + 1)))

delete\_ferz(i, j); /\* для отката (удаления ферзя на предыд. строке еще для перехода на строку выше \*/

}

if (result==true)

break; // досрочный выход из цикла после постановки ферзя на посл. строке доски

}

return result;

}

void set\_ferz(int i, int j)

{

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

doska[x][j]++; // По вертикали

doska[i][x]++; // По горизонтали

int dia = j - i + x;

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

doska[x][dia]++;

}

dia = j + i - x;

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

doska[x][dia]++;

doska[i][j] = -1; // Ставим ферзя

}

}

10

void delete\_ferz(int i, int j)

{

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

doska[x][j]--; // По вертикали

doska[i][x]--; // По горизонтали

int dia = j - i + x;

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

doska[x][dia]--;

}

dia = j + i - x;

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

doska[x][dia]--;

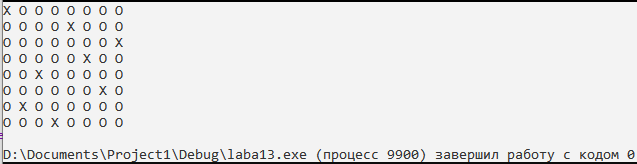
doska[i][j] = 0;

}

}

11

**Скриншоты**



12