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: «Основы алгоритмизации и программирования»

Тема: Рекурсивные функции. Задача о расстановке восьми ферзей

Выполнил:

студент группы РИС-20-2б

Пономарев Егор Витальевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Разработать программу используя метод возвратом.

**Постановка задачи**

Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого

**Анализ задачи**

Необходимые действия:

1. Создать матрицу для хранения количества ферзей, которые бьют каждую ячейку матрицы;
2. Создать функцию, которая будет пытаться поставить ферзя на i-тый столбец, если ей это удастся, то она будет возвращать истину, в противном случае – ложь;
3. В случае, если ячейка не свободна, то программа должна попытаться расположить ферзя в следующем столбце;
4. Если и в последующем случае не удалось поставить ферзя, то мы должны убрать данного ферзя с доски и уменьшить количество ячеек, которые находятся под боем;
5. Работа программы прекращается, когда все ферзи расставлены.

Создание матрицы, которая хранит количество ферзей, которые бьют каждую ячейку матрицы (8 строк и 8 столбцов):

int board[8][8];

В функции main доска отчищается, то есть заполняется нулями путем присваивания элементу матрицы к ноль:

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

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

board[i][j] = 0;

После чего вызывается функция tryQueen, которая пытается поставить ферзя на i-ый столбец доски:

tryQueen(0);

В теле функции элементы просматриваются начиная с первого столбца (0) и до последнего (7).

bool tryQueen(int i)

{

bool result = false;

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

{

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

{

setQueen(i, j);

if (i == 7) оооэ

{

result = true;

}

else if (!(result = tryQueen(i + 1)))

{

resetQueen(i, j);

}

}

if (result) break;

}

return result;

}

В этой функции инициализируется переменная, которая будет принимать значение true, если удалось поставить ферзя, и false в противном случае.

bool result = false;

Начинается проверка всех строк, и если рассматриваемую ячейку никто не бьет, то туда ставится ферзь, вызывается функция setQueen.

for (int j = 0; j < 8; j++) //проверяем все строки

{

if (board[i][j] == 0) //ячейку никто не бьет, можем поставить ферзя

{

setQueen(i, j);

В функции setQueen происходит процесс постановки ферзя. В функцию мы передаем координаты ферзя, затем по проходим по элементам по вертикали и по горизонтали увеличивая каждый элемент на 1, помимо этого программа также должна проходить по элементам диагонали(главной и побочной).

void setQueen(int i, int j) //функция, расставляющая ферзей

{

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

{

board[x][j]++;

board[i][x]++;

int d;

d = j - i + x;

if (d >= 0 && d < 8) board[x][d]++;

d = j + i - x;

if (d >= 0 && d < 8) board[x][d]++;

}

board[i][j] = -1; //значение, обозначающее, что там стоит ферзь

}

Если по окончании работы функции i равна 7, значит функция дошла до последнего столбца и все ферзи расставлены.

if (i == 7) //столбец равен 7, значит все ферзи расставлены

{

result = true;

}

В случае, если дойти до последнего столбца не удалось, то вызывается функция resetQueen, которая убирает ферзя, в теле функции происходят те же самые действия, которые происходили в функции setQueen только все наоборот.

void resetQueen(int i, int j) //функция, убирающаяя ферзей

{

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

{

board[x][j]--;

board[i][x]--;

int d;

d = j - i + x;

if (d >= 0 && d < 8) board[x][d]--;

d = j + i - x;

if (d >= 0 && d < 8) board[x][d]--;

}

board[i][j] = 0;

}

Работа прекращается если переменная result вернула значение true, что означает, что все ферзи расставлены успешно.

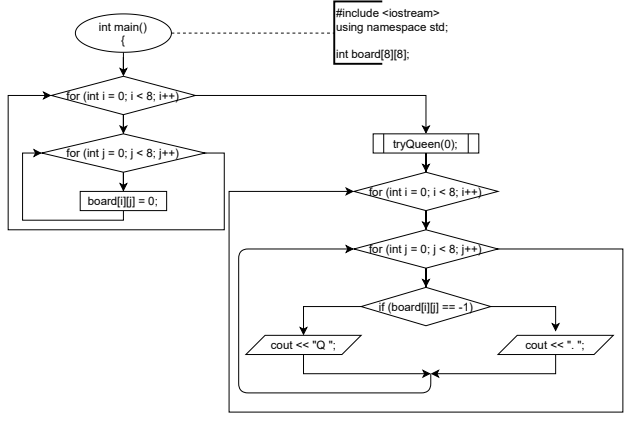
if (result) break; //если удалось поставить, то можно прекращать работу цикла

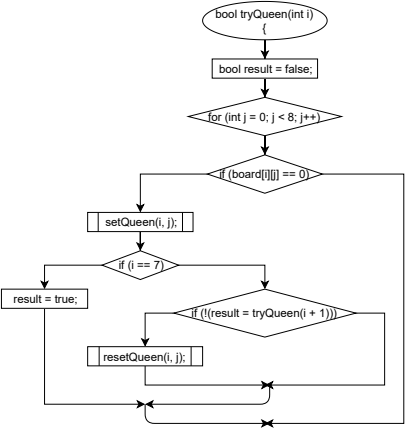
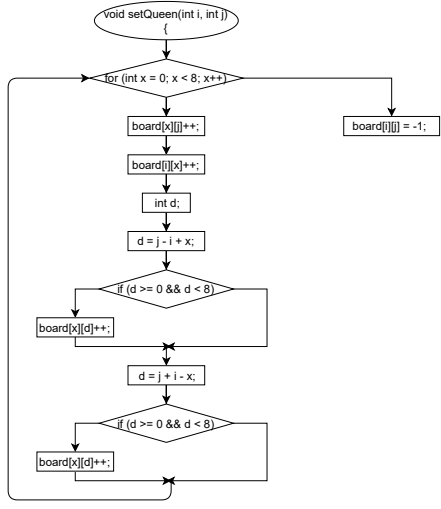
}

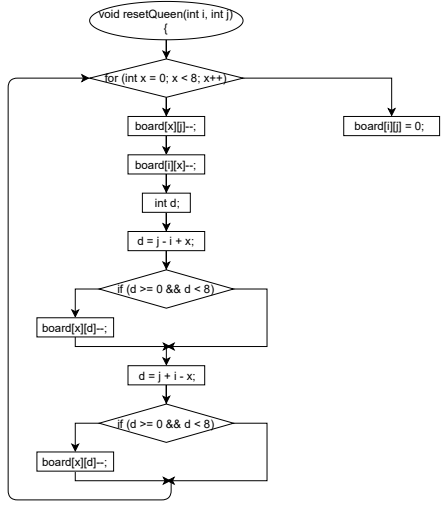
return result;

}

**Блок-схемы**



****



**Код C++**

#include <iostream>

using namespace std;

int board[8][8];

void setQueen(int i, int j)

{

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

{

board[x][j]++;

board[i][x]++;

int d;

d = j - i + x;

if (d >= 0 && d < 8) board[x][d]++;

d = j + i - x;

if (d >= 0 && d < 8) board[x][d]++;

}

board[i][j] = -1;

}

void resetQueen(int i, int j)

{

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

{

board[x][j]--;

board[i][x]--;

int d;

d = j - i + x;

if (d >= 0 && d < 8) board[x][d]--;

d = j + i - x;

if (d >= 0 && d < 8) board[x][d]--;

}

board[i][j] = 0;

}

bool tryQueen(int i)

{

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)))

{

resetQueen(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;

tryQueen(0);

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

{

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

{

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

else cout << ". ";

}

cout << endl;

}

}

**Результаты выполнения**

