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

Вариант 14

Выполнил:

Студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цели работы**

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

**Постановка задачи**

1. Сформировать двумерный массив, значения которого будут использоваться для постановки ферзи на «клетку шахматной доски»;
2. Сформировать функцию, определяющую «свободные клетки»;
3. Сформировать функцию, устанавливающую «ферзь» на свободные позиции;
4. Сформировать функцию, отменяющую последнее действие.

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Сформировать двумерный массив:   
      int doska[8][8];
   2. Создать функцию, проверяющую, свободно ли текущее место:  
      void GF(int stroka){

bool f=false;

if (was[7]<0)

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

if (doska[stroka][i]==0&&i>was[stroka]){

QSet(stroka,i);

f=true;

was[stroka]=i;

if (was[7]<0) GF(++stroka);

}

}

if (!f){

QDel(stroka);

GF(--stroka);

}

}

* 1. Создать функцию, устанавливающую ферзь на свободную позицию;

void QSet(int stroka, int stolbec){

doska[stroka][stolbec]=-1;

int str = stroka-1, stl=stolbec-1;

while (str>=0&&stl>=0) doska[str--][stl--]++;

str = stroka-1;stl=stolbec+1;

while (str>=0&&stl<8) doska[str--][stl++]++;

str = stroka+1;stl=stolbec-1;

while (str<8&&stl>=0) doska[str++][stl--]++;

str = stroka+1;stl=stolbec+1;

while (str<8&&stl<8) doska[str++][stl++]++;

for (int e=0;e<8;e++) if (doska[stroka][e]!=-1) doska[stroka][e]++;

for (int e=0;e<8;e++) if (doska[e][stolbec]!=-1) doska[e][stolbec]++;

itd++;

cout << itd << "-я итерация установки ферзи. " << endl;

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

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

if (doska [j][k]==-1) cout << doska[j][k];

else cout << " " << doska[j][k];

}

cout << endl;

}

cout << endl;

}

1. В ходе работы был использованы типы данных:
   1. int:  
      int itd=0,itu=0;
2. Данные для решения задачи представлены в виде:
   1. Целочисленных переменных:  
      int itd=0,itu=0;;
3. При решении поставленных задач использовались циклы:
   1. Для проверки свободных позиций цикл for:

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

if (doska[stroka][i]==0&&i>was[stroka]){

QSet(stroka,i);

f=true;

was[stroka]=i;

if (was[7]<0) GF(++stroka);

}

}

* 1. Для инкремента или декремента значений цикл while:

while (str>=0&&stl>=0) doska[str--][stl--]++;

while (str>=0&&stl>=0) doska[str--][stl--]--;

* 1. Для вывода значений позиций шахматной доски цикл for:

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

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

if (doska [j][k]==-1) cout << doska[j][k];

else cout << " " << doska[j][k];

}

cout << endl;

}

**Полный код программы**

#include <iostream>

using namespace std;

int doska[8][8], was[8], itd=0,itu=0;

void Set(){

const int fon = system("Color F0");

system("chcp 1251>nul");

setlocale (LC\_ALL, "Russian");

}

void QSet(int,int);

void QDel(int);

void GF(int stroka){

bool f=false;

if (was[7]<0)

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

if (doska[stroka][i]==0&&i>was[stroka]){

QSet(stroka,i);

f=true;

was[stroka]=i;

if (was[7]<0) GF(++stroka);

}

}

if (!f){

QDel(stroka);

GF(--stroka);

}

}

void QSet(int stroka, int stolbec){

doska[stroka][stolbec]=-1;

int str = stroka-1, stl=stolbec-1;

while (str>=0&&stl>=0) doska[str--][stl--]++;

str = stroka-1;stl=stolbec+1;

while (str>=0&&stl<8) doska[str--][stl++]++;

str = stroka+1;stl=stolbec-1;

while (str<8&&stl>=0) doska[str++][stl--]++;

str = stroka+1;stl=stolbec+1;

while (str<8&&stl<8) doska[str++][stl++]++;

for (int e=0;e<8;e++) if (doska[stroka][e]!=-1) doska[stroka][e]++;

for (int e=0;e<8;e++) if (doska[e][stolbec]!=-1) doska[e][stolbec]++;

itd++;

cout << itd << "-ÿ èòåðàöèÿ óñòàíîâêè ôåðçè. " << endl;

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

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

if (doska [j][k]==-1) cout << doska[j][k];

else cout << " " << doska[j][k];

}

cout << endl;

}

cout << endl;

}

void QDel(int stroka){

int str,stl;

was[stroka--]=-1;

str=stroka-1;stl=was[stroka]-1;

while (str>=0&&stl>=0) doska[str--][stl--]--;

str = stroka-1;stl=was[stroka]+1;

while (str>=0&&stl<8) doska[str--][stl++]--;

str = stroka+1;stl=was[stroka]-1;

while (str<8&&stl>=0) doska[str++][stl--]--;

str = stroka+1;stl=was[stroka]+1;

while (str<8&&stl<8) doska[str++][stl++]--;

for (int e=0;e<8;e++) if (doska[stroka][e]!=-1) doska[stroka][e]--;

for (int e=0;e<8;e++) if (doska[e][was[stroka]]!=-1) doska[e][was[stroka]]--;

doska[stroka][was[stroka]]=0;

itu++;

cout << itu << "-ÿ èòåðàöèÿ âîçâðàòà. " << endl;

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

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

if (doska [j][k]==-1) cout << doska[j][k];

else cout << " " << doska[j][k];

}

cout << endl;

}

cout << endl;

}

int main(){

Set();

for (int i=0;i<8;i++) was[i]=-1;

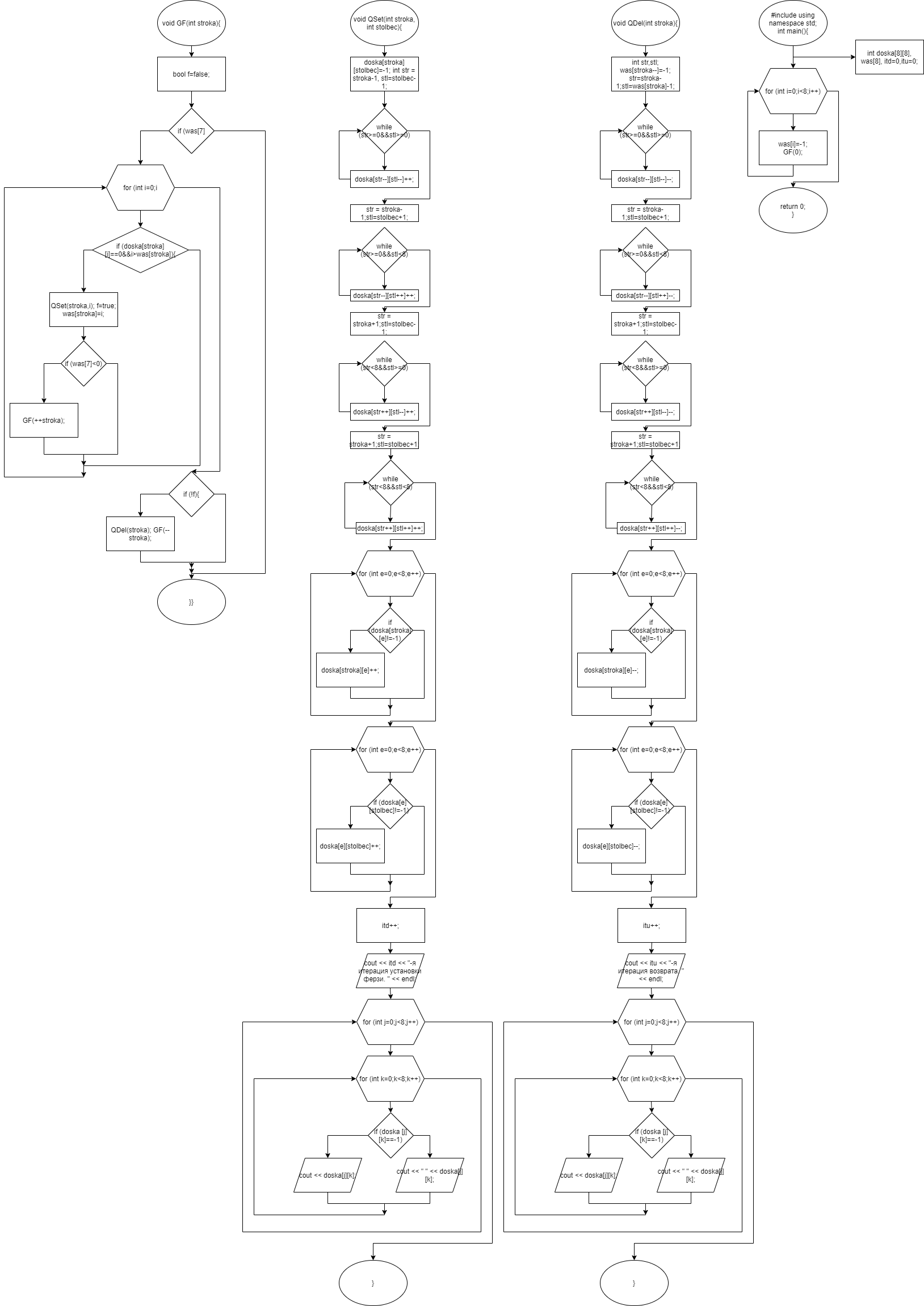
GF(0);

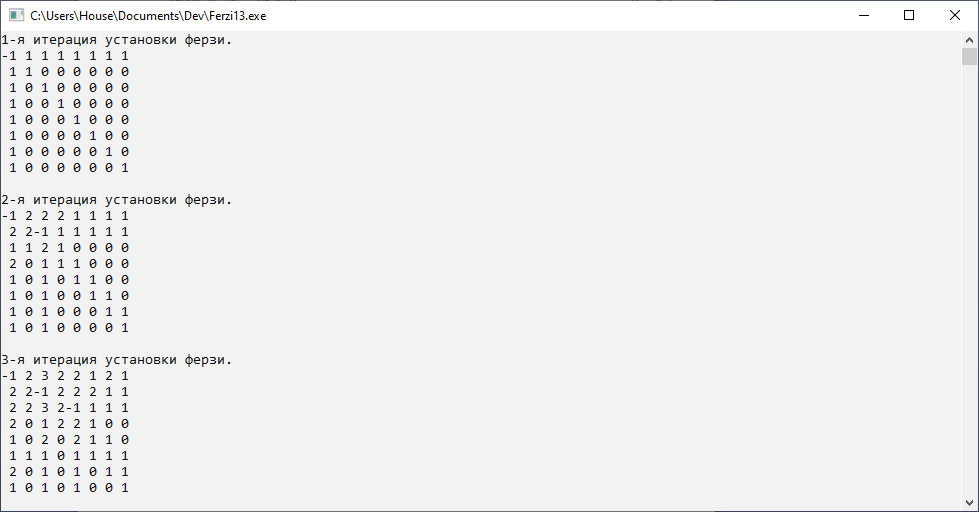
system("pause");

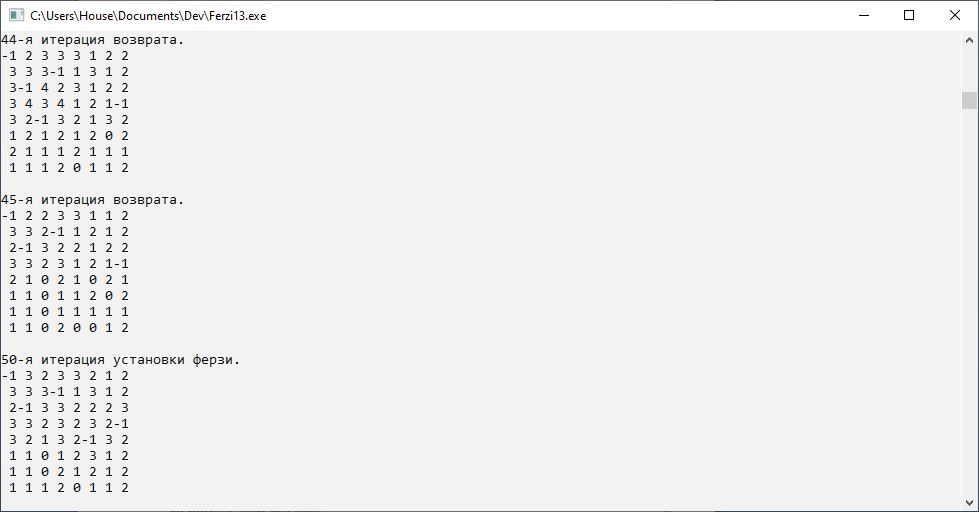
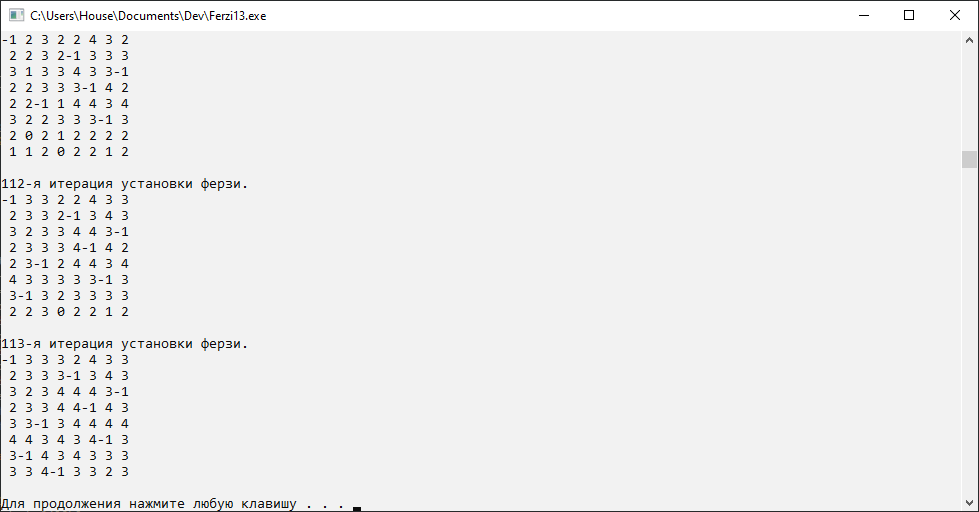
return 0;

}

**Блок-схема**



**Результаты тестов**

****