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: “Задача о 8 ферзях"

**Вар.20**

Выполнил работу

студент группы ИВТ-20-2Б

Кузнецов Н.Д

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

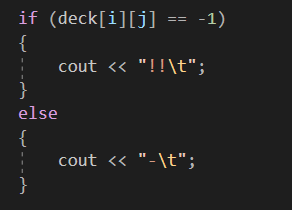
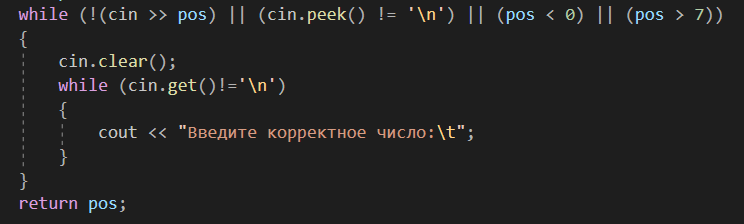
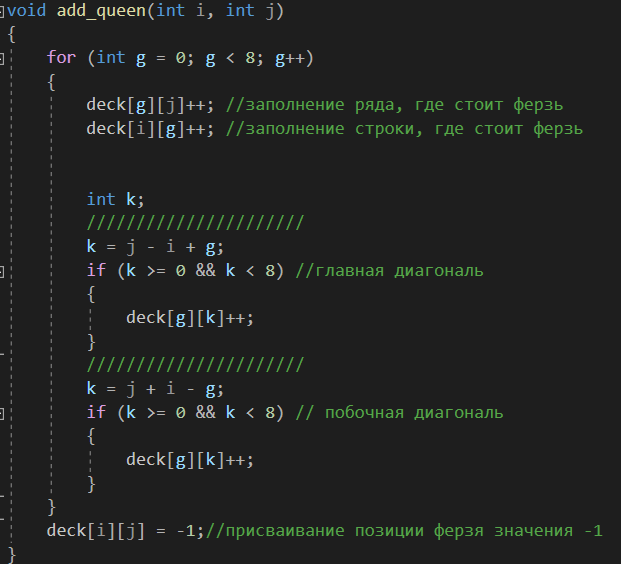
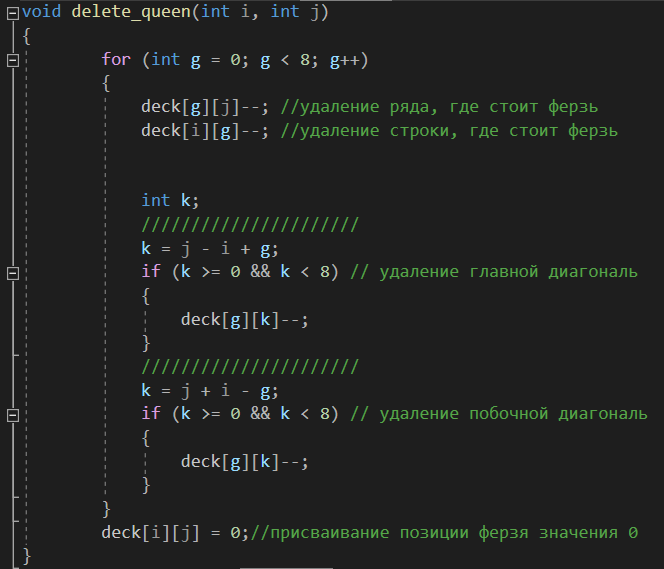
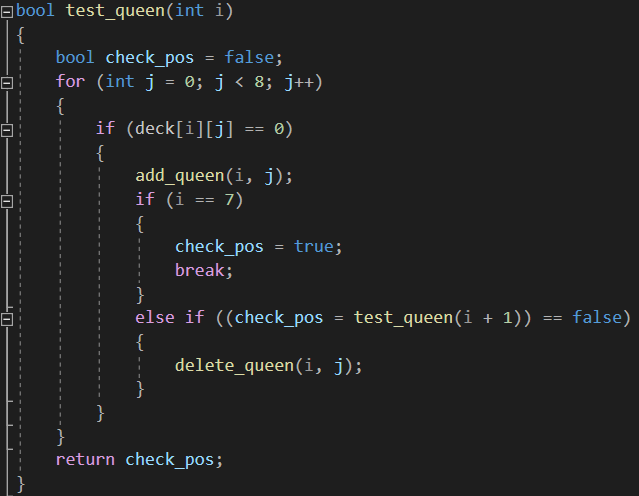
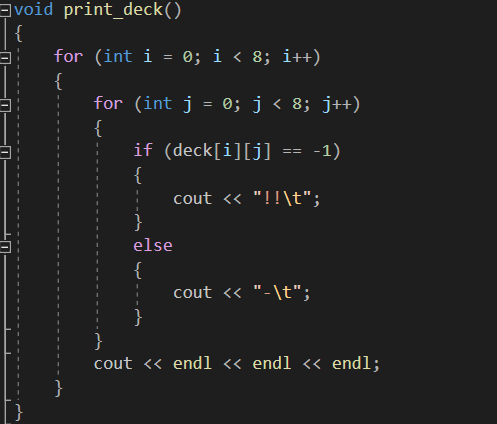
1) Получить практические навыки работы с поиском c возвратом;  
Для достижения поставленной цели необходимо решить следующие задачи:

* Разработать функцию установки ферзя.
* Разработать функцию удаления ферзя.
* Разработать функцию нахождения расстановки 8 ферзей.
* Разработать функцию вывода доски.
* Разработать программу.

**Постановка задачи**

Дана шахматная доска 8x8. Расставить ферзей так, чтобы любые два ферзя не били друг друга.

**Анализ задачи**

1. Для решения задачи используются типы данных **int**.
   1. В качестве шахматного поля выступает глобальный двумерный статический массив (**deck**) типа **int**. Он хранит в себе информацию о свободных клетках (0), клетках, занятых ферзями (-1), клетках, находящихся под боем (число большее, чем нуль).  
      
   2. Функция добавления и удаления ферзя принимают координаты самого ферзя (i, j) и не возвращают значений.  
      
   3. Функция проверки следующей строки на возможность поставить ферзя возвращает значение типа **bool**. В качетсве аргумента функции выступает текущая строка (i).  
      
   4. Функция вывода решения с доски не возвращает значения, список ее параметров пуст.  
      
   5. Функция получения позиции от пользователя возвращает тип данных **int**, не принимает значений.  
      
2. Для решения задачи необходимо:
   1. Разработать функцию установки ферзя (add\_queen).
   2. Разработать функцию удаления ферзя с позиции (delete\_queen).
   3. Разработать функцию, проверяющую следующую строку на возможность поставить ферзя (test\_queen).
   4. Разработать функцию, выводящую доску пользователю (print\_deck).
   5. Разработать функцию, которая проверяет на ввод корректность позиции первого ферзя в первой строке (GetPos).
3. Для решения задачи данные были представлены в следующем виде:
   1. Поле шахматной доски представлено в виде матрицы 8x8 (см.п.1.1).
4. Для операций ввода и вывода использовались следующие операторы и функции:
   1. Функция print\_deck использует оператор cout.  
      
   2. Функция GetPos использует операторы cin, cout.  
      
5. Поставленные задачи будут решены следующими действиями:
   1. Функция add\_queen заполняет клетки главной и побочной диагонали атаки ферзя, его расположение, клетки столбца и строки. Это реализовано с помощью цикла for. Заполнение ряда происходит с помощью индексации deck[g][j], где j-ряд, в котором стоит ферзь, g - управляющая переменная цикла. Значение в этих ячейках инкрементируется. Аналогично заполнению ряда происходит заполнение строки, за одним лишь исключением - индексация deck[i][g]. Для заполнения диагоналей используются операторы ветвления if, с условием k >= 0 && k < 8, где k для главной диагонали рассчитывается по формуле k = j - i + g, а для побочной k = j + i – g. После выхода из цикла ячейке с индексом i, j присваивается значение -1.  
      
   2. Аналогично п.5.1 функция delete\_queen удаляет ферзя, за исключением того, что значение клеток, находящихся под боем, декрементируется, значению ячейки с индексами i, j присваивается значение **0**.  
      
   3. Функция test\_queen использует переменную типа bool, значение которой изначально равно false. Функция работает по принципу рекурсии. Перебор клеток осуществляется с помощью цикла for, если найдена клетка со значением 0, то вызывается функция add\_queen, которая ставит ферзя в позицию i строки j столбца (j - управляющая переменная цикла). Если значение i равно 7, то функция прекращает свою работу (все ферзи расставлены). Иначе, вызывается функция test\_queen с параметром i+1. Если функция возвращает значение false, то ферзь удаляется, и перебор строки продолжается.  
      
   4. Функция print\_deck “проходится” по доске с помощью вложенных циклов for, если элемент с индексами [i][j] равен -1, то на экран выводится “!!”, иначе “-”.  
      

**Код программы**

#include <iostream>

using namespace std;

int deck[8][8] = {};

void add\_queen(int i, int j)

{

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

{

deck[g][j]++; //заполнение ряда, где стоит ферзь

deck[i][g]++; //заполнение строки, где стоит ферзь

int k;

//////////////////////

k = j - i + g;

if (k >= 0 && k < 8) //главная диагональ

{

deck[g][k]++;

}

//////////////////////

k = j + i - g;

if (k >= 0 && k < 8) // побочная диагональ

{

deck[g][k]++;

}

}

deck[i][j] = -1;//присваивание позиции ферзя значения -1

}

void delete\_queen(int i, int j)

{

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

{

deck[g][j]--; //удаление ряда, где стоит ферзь

deck[i][g]--; //удаление строки, где стоит ферзь

int k;

//////////////////////

k = j - i + g;

if (k >= 0 && k < 8) // удаление главной диагональ

{

deck[g][k]--;

}

//////////////////////

k = j + i - g;

if (k >= 0 && k < 8) // удаление побочной диагональ

{

deck[g][k]--;

}

}

deck[i][j] = 0;//присваивание позиции ферзя значения 0

}

bool test\_queen(int i)

{

bool check\_pos = false;

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

{

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

{

add\_queen(i, j);

if (i == 7)

{

check\_pos = true;

break;

}

else if ((check\_pos = test\_queen(i + 1)) == false)

{

delete\_queen(i, j);

}

}

}

return check\_pos;

}

void print\_deck()

{

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

{

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

{

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

{

cout << "!!\t";

}

else

{

cout << "-\t";

}

}

cout << endl << endl << endl;

}

}

int GetPos()

{

int pos;

while (!(cin >> pos) || (cin.peek() != '\n') || (pos < 0) || (pos > 7))

{

cin.clear();

while (cin.get()!='\n')

{

cout << "Введите корректное число:\t";

}

}

return pos;

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Введите позицию в первой строке (от 0 до 7):\t";

int PosX = GetPos();

cout << "===================================================" << endl;

add\_queen(0, PosX);

if (test\_queen(1))

{

print\_deck();

}

else

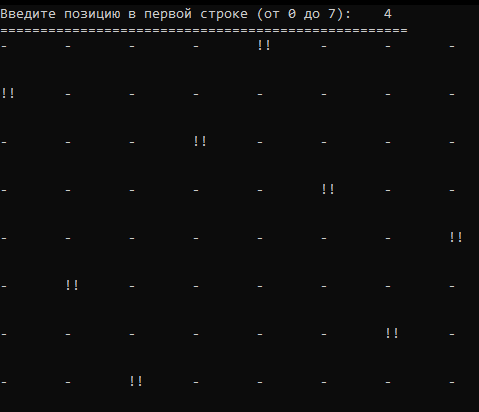
{

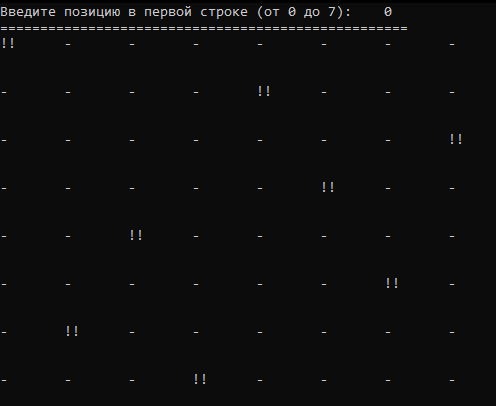
cout << "Решения для данной позиции в 1 строке нет!" << endl;

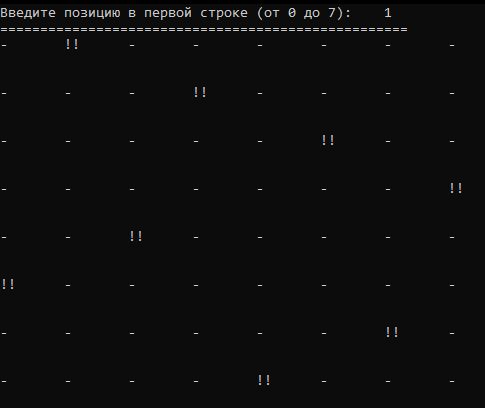
}

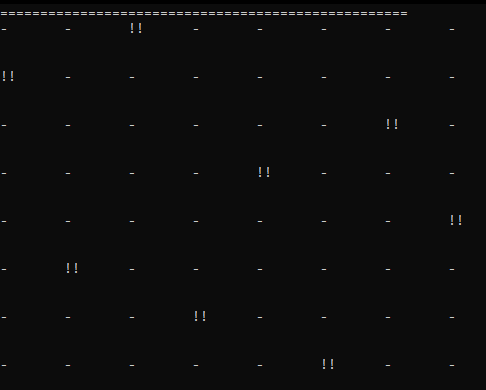
}

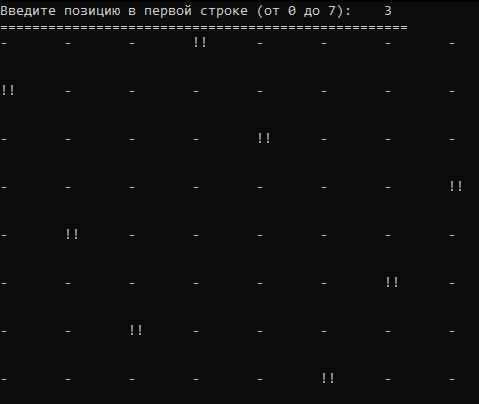
**Работа программы**

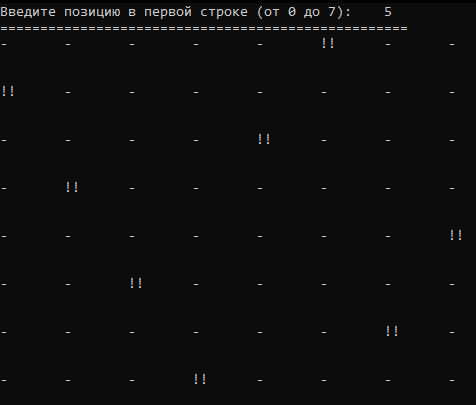


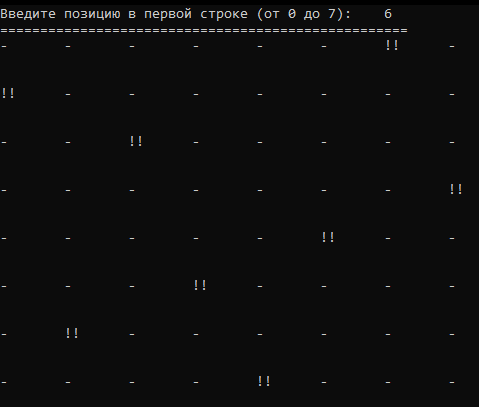


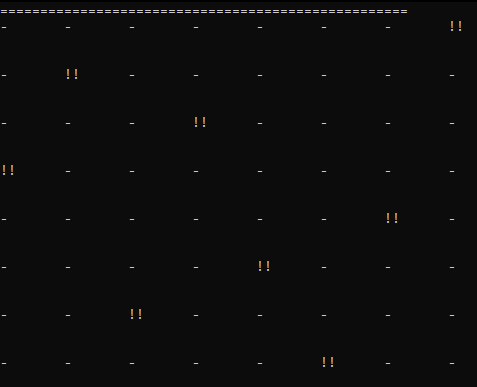












**Блок-схема**

