Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**Тема:** «Задача о восьми ферзях»

Семестр: 2

|  |  |  |
| --- | --- | --- |
|  |  | Выполнил: |
|  |  | Студент группы ИВТ-23-2б |
|  |  | Ходырев Василий Алексеевич |
|  |  |  |
|  |  | (дата, подпись) |
|  |  |  |
|  |  | Проверила: |
|  |  | Доцент кафедры ИТАС |
|  |  | О.А. Полякова |
|  |  |  |
|  |  | (дата, подпись) |

Пермь 2024

Анализ решения задачи:

Ферзь может ходить в любом направлении по горизонтали, вертикали, диагонали и на любое количество клеток, рубит он так же, как ходит.

Чтобы ферзи друг друга не «били», на каждой строке, диагонали и каждом столбце должен находиться один ферзь.

Для расстановки ферзей требуется:

1. Поставить первого ферзя на позицию а1 (первая клетка первой строки).
2. Перейти на следующую строку и поставить ферзя так, чтобы первый ферзь его не бил.
3. Если на какой-либо строке поставить ферзя невозможно(так, чтобы они не «били» друг друга), то возвращаемся на предыдущую строку и ставим ферзя на следующую клетку строки.
4. Повторяем пункты 2 и 3, пока не расставим всех ферзей.

Алгоритм решения:  
1. Необходимо реализовать две функции: поставить ферзя и убрать ферзя. Первая будет ставить в заданную клетку ферзя, а во все остальные отмечать, что они находятся под боем. Вторая аналогично будет убирать ферзя и клетки шахматной доски, находящиеся под боем.

2. Помимо этого должна быть функция, которая выбирает, куда поставить ферзей, она поочередно проходит строки и ставит ферзей так, чтобы они находились не под боем(ферзи не должны находиться в одной строке, в одном столбце и на одной диагонали). Но если на какой-то строке функция уже не может поставить ферзя, то она возвращается на шаг назад, убирает предыдущего ферзя и пытается поставить его на другое место. И так до тех пор, пока все ферзи не займут свои места.

Изначально определен двумерный массив board, который обозначает шахматную доску 8 на 8.

В главной функции значение всех элементов двумерного массива приравнивается к 0. Вызывается функция tryQueen, в качестве параметра передается значение 0 – первая строка шахматной доски. Когда отмечены все возможные расстановки ферзей, выводится схема шахматной доски, при этом все значения элементов, равные -1, отмечаются буквой «Q»(ферзи).

Шахматная доска соответствует двумерный массив размерностью 8 на 8, в котором будут расставляться ферзи.

Изначально массив board заполнен 0.

Функция setQueen ставит на позицию [i][j] ферзя и отмечает те позиции, которые данный ферзь «бьет».

Поставить ферзя – значит проинициализировать элемент с индексами [i][j] -1. Отметить позиции, которые данный ферзь «бьет» - значит прибавить 1 к значениям элементов, которые находятся под «боем».

Функция resetQueen убирает с позиции [i][j] ферзя и убирает отметки с тех позиций, которые данный ферзь «бил».

Убрать ферзя – значит проинициализировать элемент с индексами [i][j] 0. Отметить позиции, которые данный ферзь «бил» - значит отнять 1 от значений элементов, которые находились под «боем».

Функция tryQueen проверяет, можно ли поставить ферзя на данную позицию.

В цикле for проверяются все элементы строки; если очередной элемент равен 0, на данную позицию ставится ферзь. Далее осуществляется переход на следующую строчку. Если в какой-либо из последующих строк ни в один столбец нельзя поставить ферзя, осуществляется переход на предыдущий шаг, и ферзь ставится уже на следующую допустимую позицию.

Код программы

#include <iostream>

using namespace std;

int board[8][8];

void setQueen(int i, int j)

{

for (int x = 0; x <= 7; ++x)

{

++board[x][j];

++board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7)

++board[x][foo];

foo = j + i - x;

if (foo >= 0 && foo <= 7)

++board[x][foo];

}

board[i][j] = -1;

}

void resetQueen(int i, int j)

{

for (int x = 0; x <= 7; ++x)

{

--board[x][j];

--board[i][x];

int foo;

foo = j - i + x;

if (foo >= 0 && foo <= 7)

--board[x][foo];

foo = j + i - x;

if (foo >= 0 && foo <= 7)

--board[x][foo];

}

board[i][j] = 0;

}

bool tryQueen(int i)

{

bool result = false;

for (int j = 0; j <= 7; ++j)

{

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

{

setQueen(i, j);

if (i == 7)//всех королев поставили

result = true;

else

{

result = tryQueen(i + 1);

if (!result)

resetQueen(i, j);

}

}

}

return result;

}

int main()

{

for (int i = 0; i <= 7; ++i)

for (int j = 0; j <= 7; ++j)

board[i][j] = 0;

tryQueen(0);

for (int i = 0; i <= 7; ++i)

{

for (int j = 0; j <= 7; ++j)

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

cout << "Q ";

else

cout << ". ";

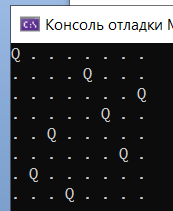
cout << "\n";

}

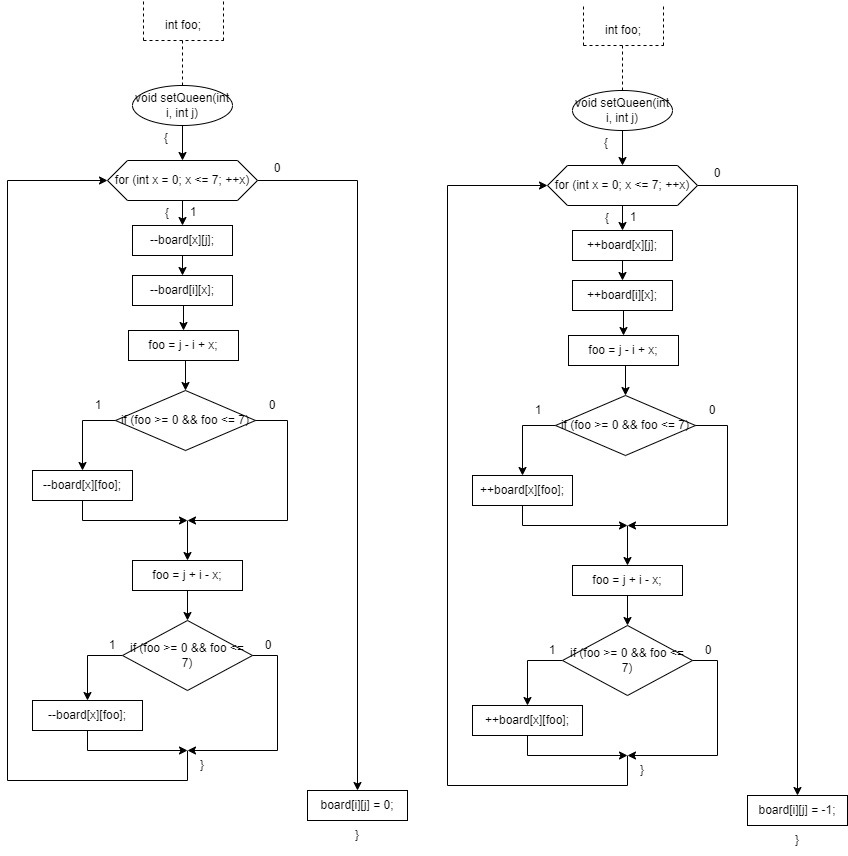
return 0;

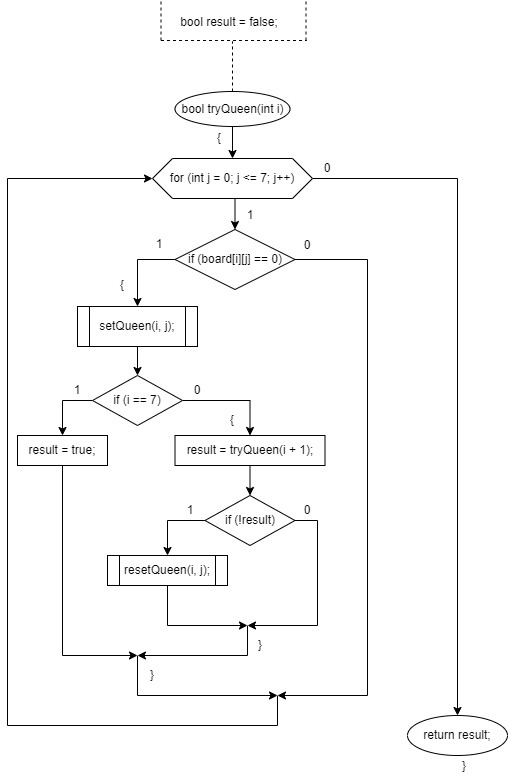
}

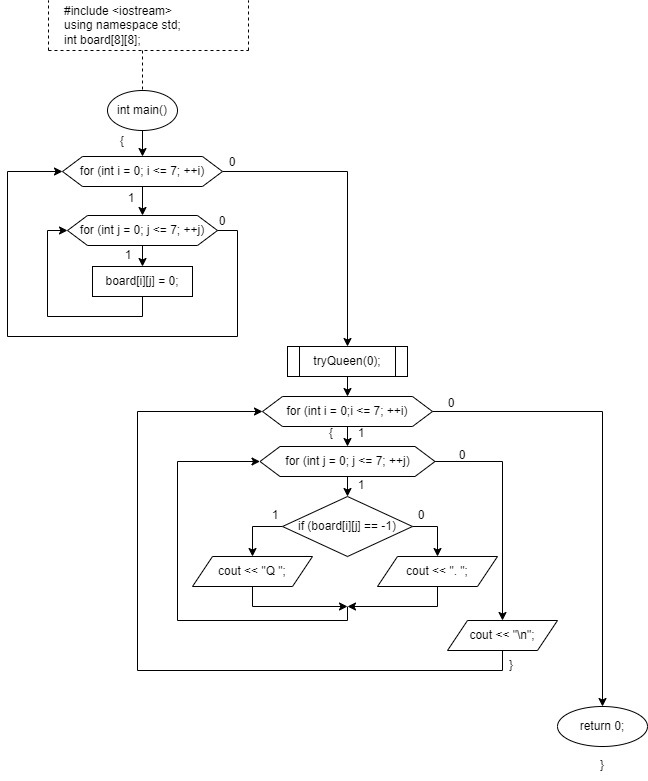
Вывод



Блок-схема







Визуализация

<https://youtu.be/-iX5OCbcKfI>

(скопируйте ссылку, чтобы посмотреть визуализацию)