Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

**“8 ферзей”**

Выполнил:

студент группы РИС-23-2б

Камалетдинов Максим Валерьевич

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Постановка задачи:**

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

Подразумевается, что ферзь бьёт все клетки, расположенные по вертикалям, горизонталям и обеим диагоналям.

**Программный код:**

#include<iostream>

#include<clocale>

using namespace std;

int board[8][8];

void putqueen(int i, int j) {

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

++board[x][j];

++board[i][x];

int m;

m = j - i + x;

if (m >= 0 && m <= 7) ++board[x][m];

m = j + i - x;

if (m >= 0 && m <= 7) ++board[x][m];

}

board[i][j] = -1;

}

void removequeen(int i, int j) {

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

--board[x][j];

--board[i][x];

int m;

m = j - i + x;

if (m >= 0 && m <= 7) --board[x][m];

m = j + i - x;

if (m >= 0 && m <= 7) --board[x][m];

}

board[i][j] = 0;

}

bool tryqueen(int i) {

bool result = false;

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

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

putqueen(i, j);

if (i == 7) result = true;

else {

result = tryqueen(i + 1);

if (result==false) removequeen(i, j);

}

}

}

return result;

}

int main(){

setlocale(LC\_ALL,"Ru");

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 << "\*";

}

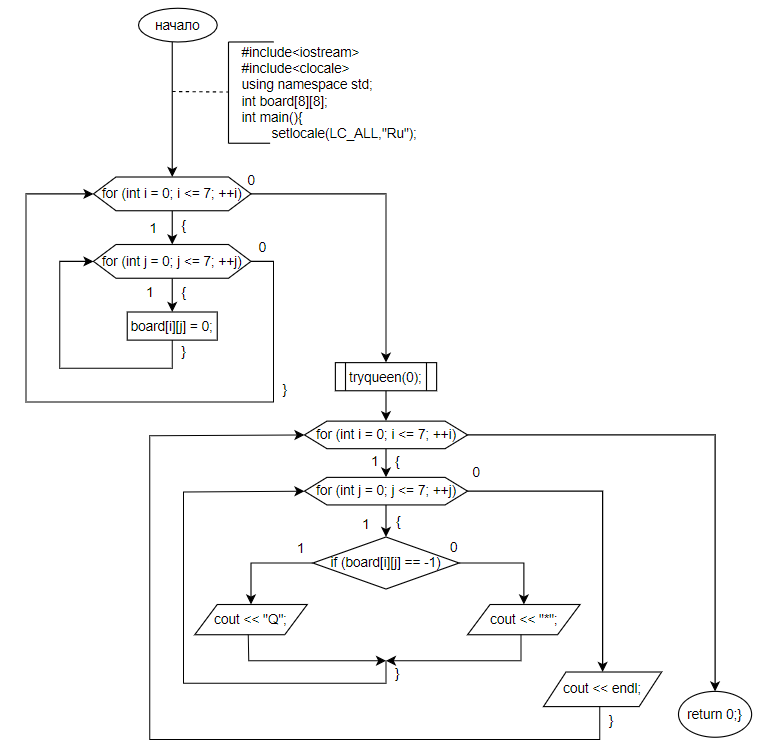
cout << endl;

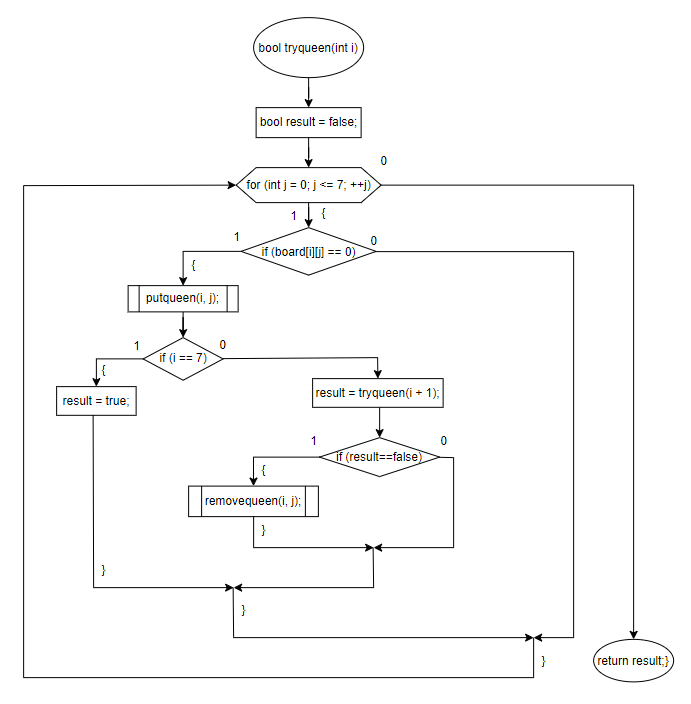
}

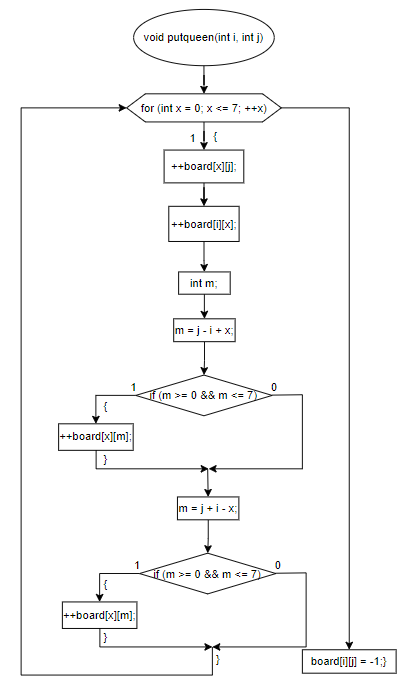
return 0;

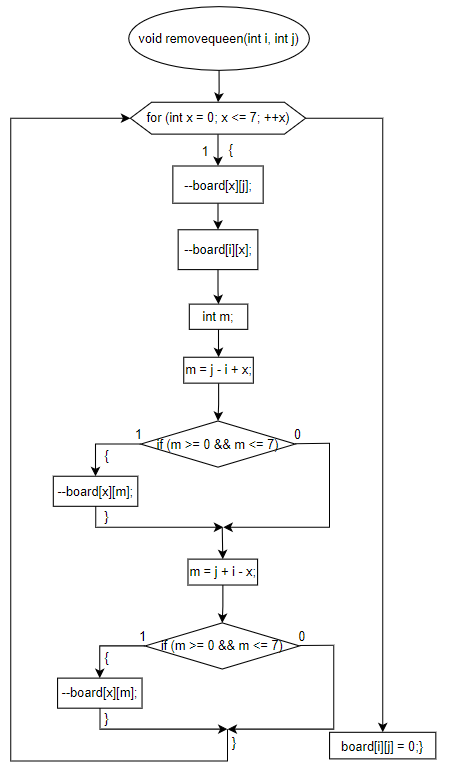
}

**Блок-схема:**

****

****

****

****

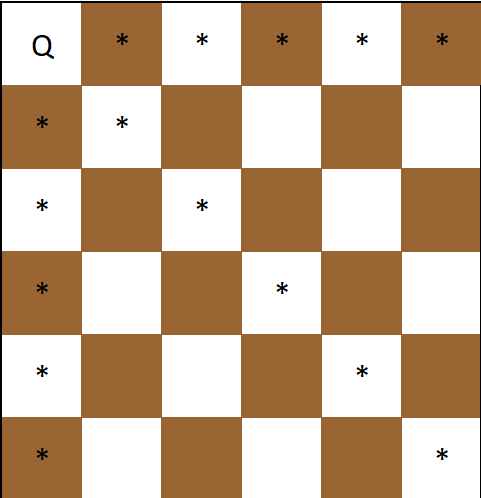
**Словесный алгоритм:**

1. Определяем двумерный массив с именем board, размером 8х8, который обозначает шахматную доску 8 на 8.
2. Создаем пустую доску, заполненную 0.
3. Заходим в функцию tryqueen(0), которая отвечает за проверку возможности постановки ферзя на какую-либо позицию(в качестве параметра передается 0, то есть первая строка шахматной строки).
   1. В функции tryqueen, функция принимает на вход индекс строки.
   2. Создается флажок result изначально равный false, который будет отвечать за конец расстановки.
   3. Создается цикл for, который осуществляет проход по всем клеткам строки и если очередной элемент равен 0, то мы заходим в функцию putqueen.
   4. Функция putqueen отвечает за постановку ферзя и отмечает те позиции, которые он бьет.
      1. Функция putqueen принимает на вход индексы свободной клетки (ту клетку, куда можно поставить ферзя)
      2. Создается цикл for для обозначения клеток, которые находятся под боем по разным направлениям. Для этого нужно во все клетки под боем добавить 1(++board[x][j];//по столбцу сверху вниз ++board[i][x];//по строке слева направо).
      3. Создается целочисленная переменная m, которая задает направления двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя.
      4. board[i][j]=-1(ферзь был поставлен)
   5. Если индекс строки i равен 7, это значит, что все ферзи были успешно расставлены и флажок result станет равен true.
   6. Иначе, result=tryqueen(i+1); Если флажок result равен false(то есть следующего ферзя нельзя поставить), то мы заходим в функцию removequeen(меняем позицию текущего).
      1. На вход функции принимаются индексы клетки в которой на данный момент стоит ферзь.
      2. Создается цикл for, который убирает все клетки, которые находятся под боем по разным направлениям. Для этого нужно из всех таких клеток под боем вычесть 1(--board[x][j];//по столбцу сверху вниз --board[i][x];//по строке слева направо).
      3. Создается целочисленная переменная m, которая задает направления двух диагоналей так, что они проходят через ферзя и их координата по строке будет симметрична относительно ферзя.
      4. board[i][j]=0(ферзь был убран)
   7. Возвращаем переменную result.
4. Создаем внешний цикл for, который отвечает за строки и внутренний for-за столбцы. Эти циклы отвечают за вывод нашей доски board в которой если значение элемента будет равно -1, то на этом месте будет выводится “Q”(ферзь), а иначе “\*”.

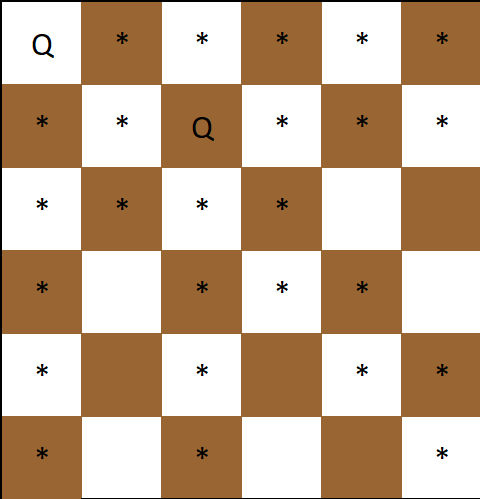
**Визуализация:**

Рассмотрим пример на доске 6х6

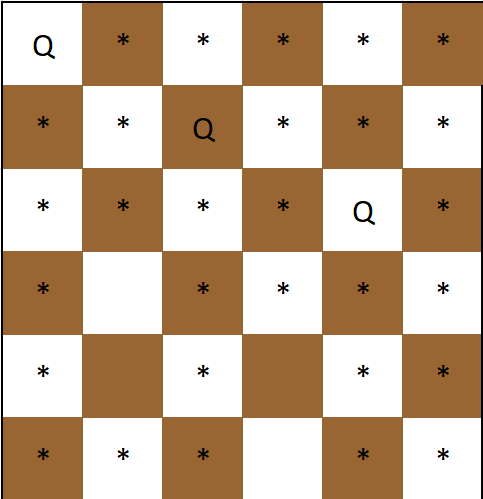
1. Ставим первого ферзя на первую пустую клетку и отмечаем клетки, которые он бьет “\*”.



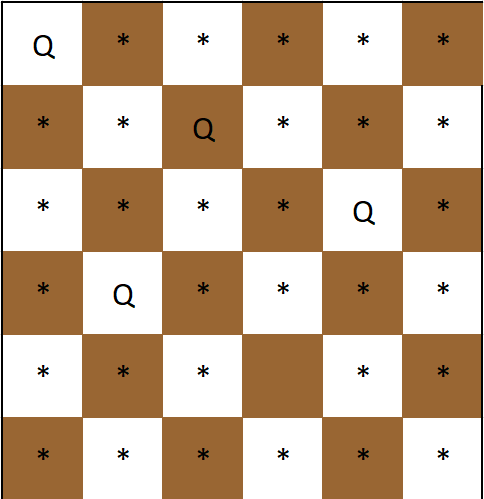
1. Переходим на вторую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



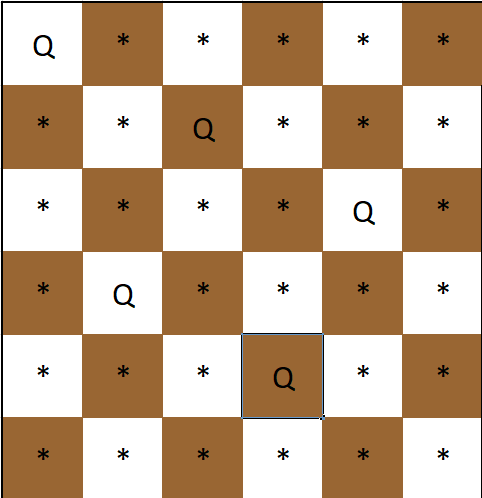
1. Переходим на третью строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



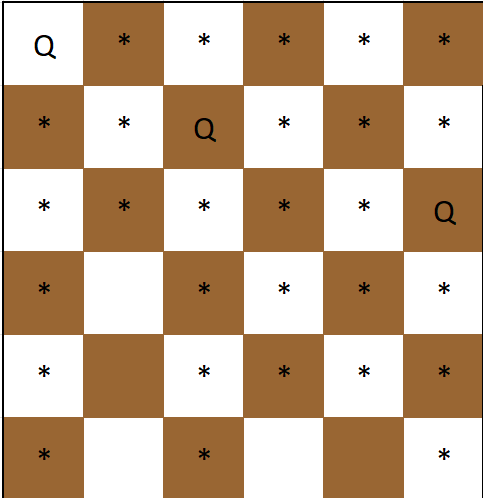
1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



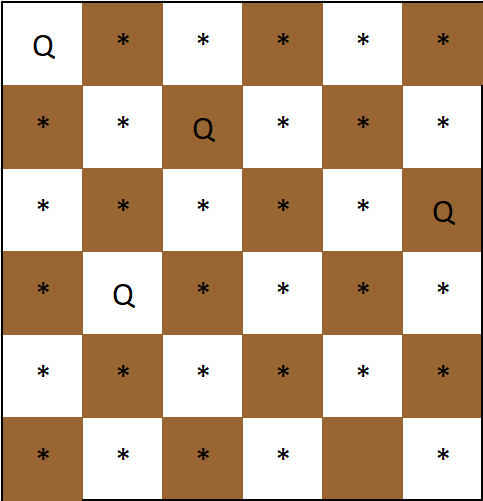
1. Переходим на пятую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



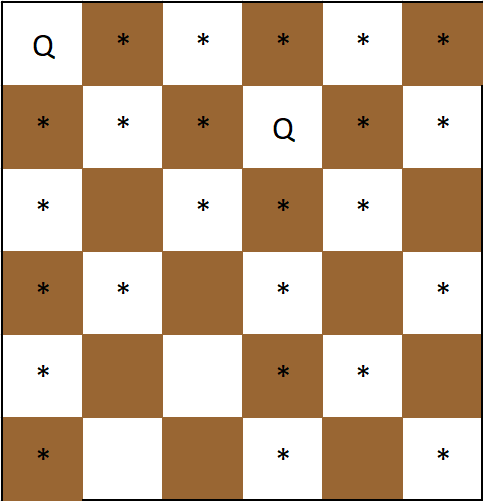
1. Замечаем, что шестого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так как 5 и 4 ферзей переставлять некуда, то вернемся к 3 ферзю и будем менять его позицию.



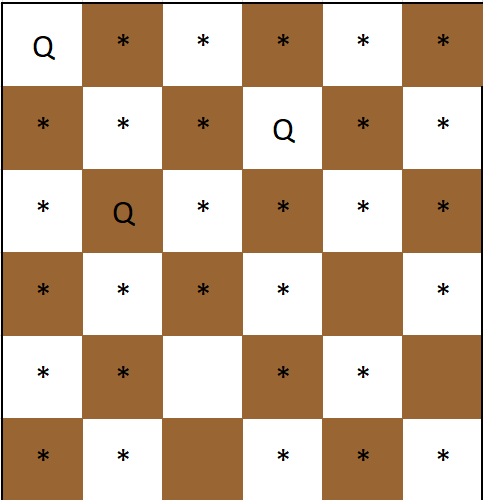
1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



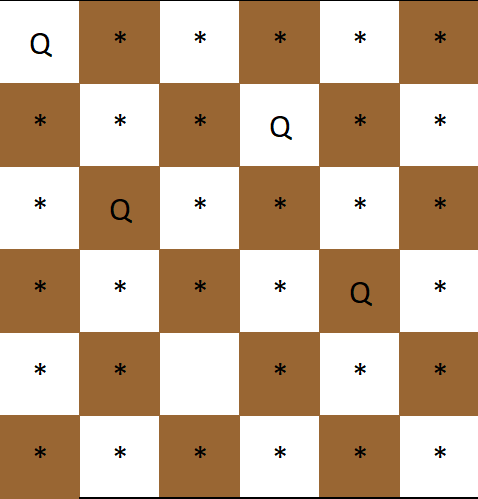
1. Замечаем, что пятого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так как 3 и 4 ферзей переставлять некуда, то вернемся ко 2 ферзю и будем менять его позицию.



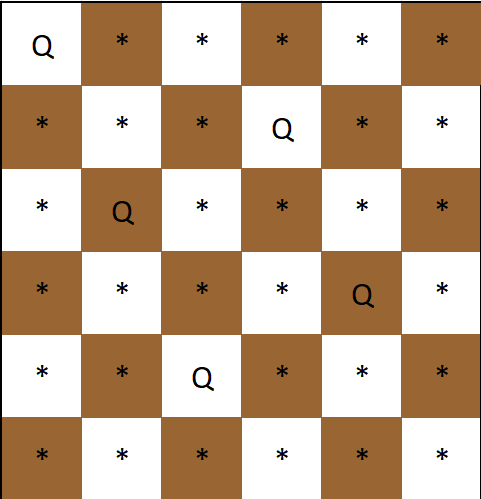
1. Переходим на третью строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



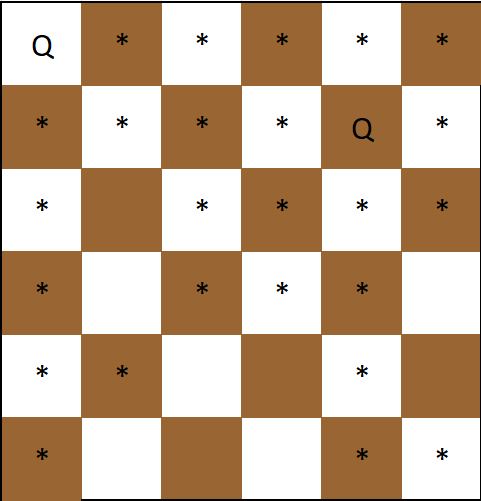
1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



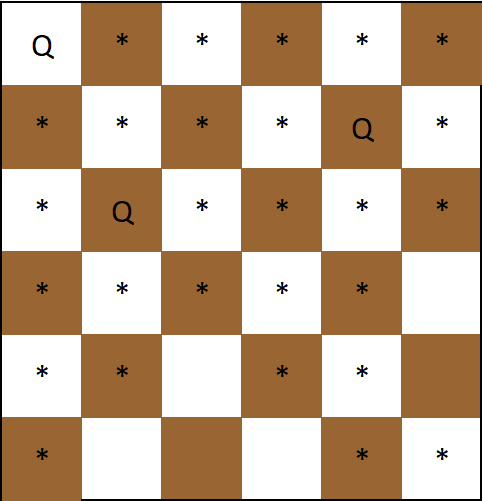
1. Переходим на пятую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



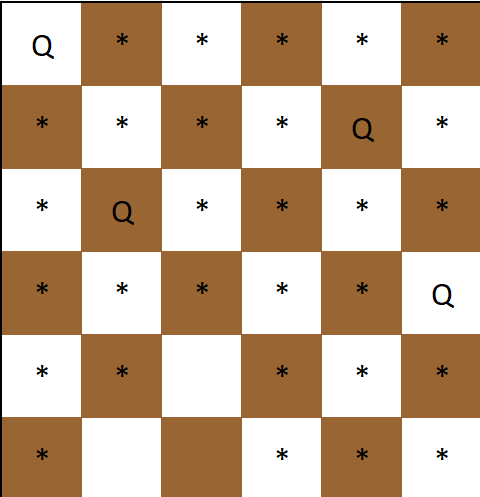
1. Замечаем, что шестого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так как 5, 4,3 ферзей переставлять некуда, то вернемся ко 2 ферзю и будем менять его позицию.



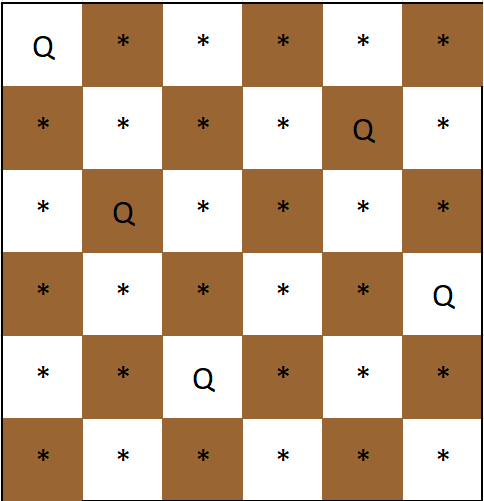
1. Переходим на третью строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



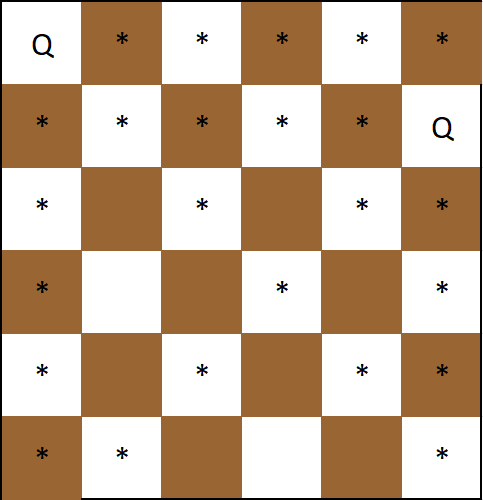
1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



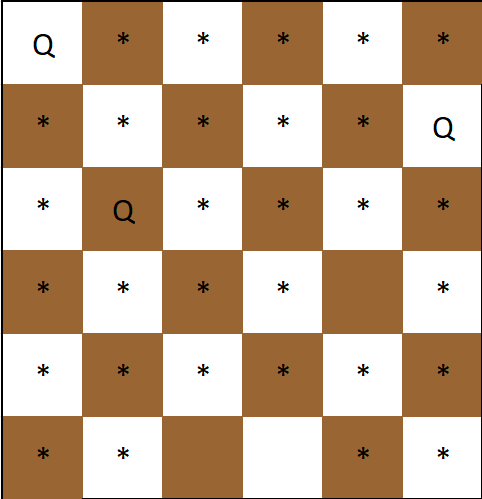
1. Переходим на пятую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



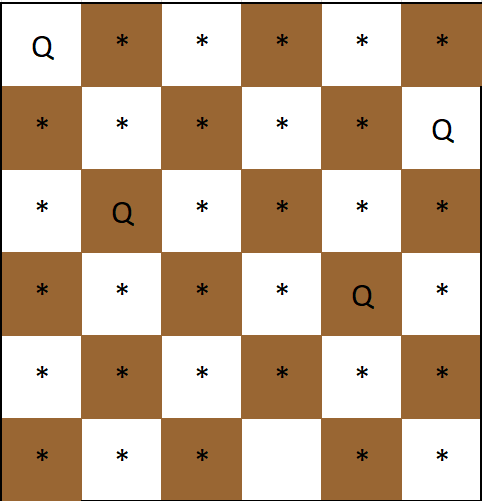
1. Замечаем, что шестого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так как 5, 4,3 ферзей переставлять некуда, то вернемся ко 2 ферзю и будем менять его позицию.



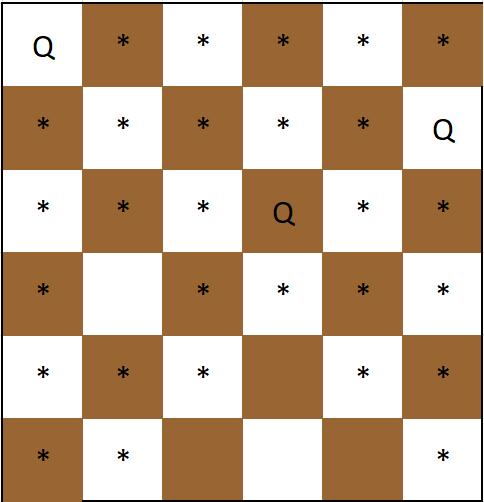
1. Переходим на третью строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



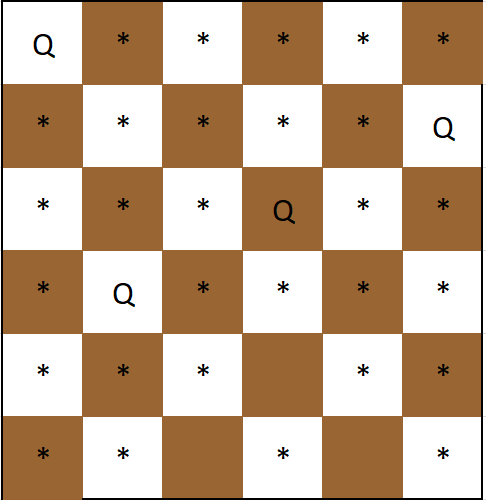
1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



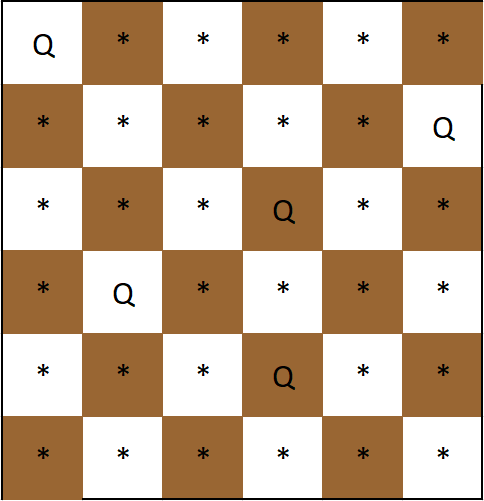
1. Замечаем, что пятого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так 4 ферзя переставлять некуда, то вернемся к 3 ферзю и будем менять его позицию.



1. Переходим на четвертую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



1. Переходим на пятую строку и ставим там ферзя на первую встретившуюся клетку, которая нам подходит (пустая, то есть никакой предыдущий поставленный ферзь её не бьет) и отмечаем “\*” клетки, которые он бьет.



1. Замечаем, что шестого ферзя некуда ставить, значит нужно переставить предыдущего ферзя, но так 5,4,3,2 ферзей переставлять некуда, то вернемся к 1 ферзю и будем менять его позицию.
2. Таким образом мы будет расставлять ферзей до момента пока они не встанут в таком положении:

