Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

Отчет по лабораторной работе

**«Восемь ферзей»**

Выполнила:

студент группы РИС-23-2б

Кладов Никита Владимирович

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Создание алгоритма для выполнения задачи**

**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.

**Словесный алгоритм:**

1. Создать функцию, которая будет проверять, является ли данная расстановка ферзей на доске правильной. Эта функция будет принимать параметр - текущий ряд.

2. Внутри функция перебирает строку и ставит ферзя, если есть “пустое” место. Если дойдет до конца строки и не найдет пустого места - убирает ферзя с предыдущего шага и ставит его в другую клетку. Если Перебор доходит последней строки и в последнюю строку ставится ферзь - то искомая комбинация расстановки найдена.

3. Создать функцию, которая будет ставить ферзя на доске и отмечать атакованные клетки. Доска является двумерным массивом, если в клетке 0 - она свободна и в нее можно поставить ферзя, если в клетке -1 - то в ней стоит ферзь, если значение в клетке больше n > 0 - то эту клетку бьют n ферзей.

4. Внутри функция ставит ферзя в клетку с координатами, найденными в функции из п. 1, а также отмечает клетки, которые бьет этот ферзь (т. е. значение клетки ++1).

5. Создать функцию, которая будет убирать ферзя с доске и отмечать атакованные этим ферзем клетки. Значение в “битых” клетках - -1.

6. Создать функцию, которая будет выводить доску в зависимости от значений. (-1 - ферзь, 0 - пустая клетка, n>0 - клетка бьется другим ферзем)

**Визуализация решения**

1. **Board()**
2. **CheckQueen(0);** 
   1. ***i=0, j=0, board[0][0] == 0***
   2. ***PlaceQeen(0, 0) // result = false***
   3. ***if (!(result = checkQueen(i + 1)))***

***2.1 checkQueen(0 + 1)***

1. ***i=1, j=0, board[1][0] == 1***
2. ***i=1, j=1, board[1][1] == 1***
3. ***i=1, j=2, board[1][2] == 0***
4. ***PlaceQeen(1, 2) // result = false***
5. ***if (!(result = checkQueen(1 + 1)))***

***2.1.1 checkQueen(1 + 1)***

1. ***i=2, j=0, board[2][0] == 2***
2. ***i=2, j=1, board[2][1] == 1***
3. ***i=2, j=2, board[2][2] == 2***
4. ***i=2, j=3, board[2][3] == 1***

***Решения нет. RemoveQueen(1, 2) и возврат на предыдущий шаг***

***2.2 checkQueen(0 + 1)***

1. ***i=1, j=0, board[1][0] == 1***
2. ***i=1, j=1, board[1][1] == 1***
3. ***i=1, j=2, board[1][2] == 0***
4. ***i=1, j=3, board[1][3] == 0***
5. ***PlaceQeen(1, 3) // result = false***
6. ***if (!(result = checkQueen(1 + 1)))***

***2.2.1 checkQueen(1 + 1)***

1. ***i=2, j=0, board[2][0] == 2***
2. ***i=2, j=1, board[2][1] == 0***
3. ***PlaceQeen(2, 1)*** 
4. ***if (!(result = checkQueen(2 + 1)))***

***2.2.1.1 checkQueen(2 + 1)***

1. ***i=3, j=0, board[3][0] == 2***
2. ***i=3, j=1, board[3][1] == 2***
3. ***i=3, j=2, board[3][2] == 1***
4. ***i=3, j=3, board[3][3] == 2***

***Решения нет. RemoveQueen(1, 3) и возврат на два шага***

* 1. ***i=1, j=0, board[1][0] == 0***
  2. ***PlaceQeen(0, 0) // result = false***
  3. ***if (!(result = checkQueen(i + 1)))***

***2.1 checkQueen(0 + 1)***

1. ***i=1, j=0, board[1][0] == 1***
2. ***i=1, j=1, board[1][1] == 1***
3. ***i=1, j=2, board[1][2] == 1***
4. ***i=1, j=3, board[1][3] == 0***
5. ***PlaceQeen(1, 3) // result = false***
6. ***if (!(result = checkQueen(1 + 1)))***

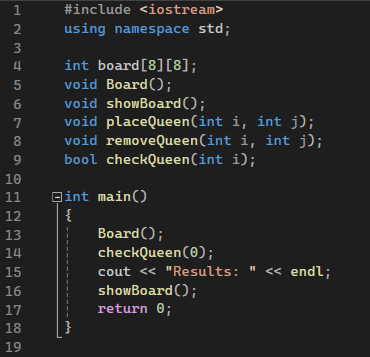
***2.1.1 checkQueen(1 + 1)***

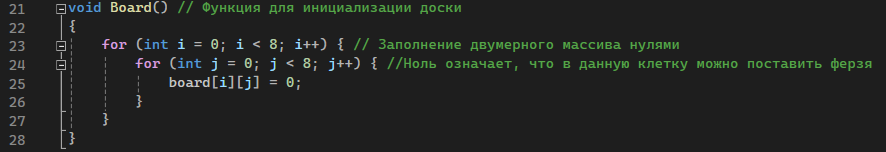
1. ***i=2, j=0, board[2][0] == 0***
2. ***PlaceQeen(2, 0)***
3. ***if (!(result = checkQueen(2 + 1)))***

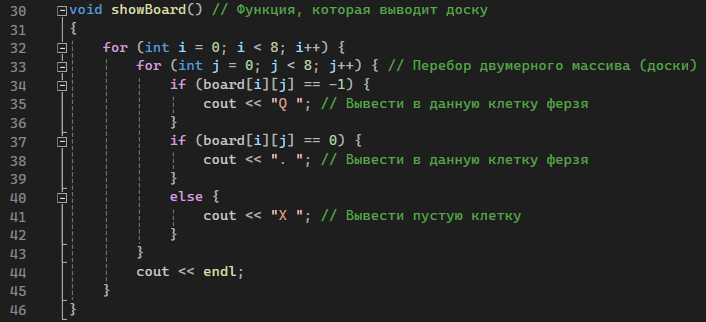
***2.2.1.1 checkQueen(2 + 1)***

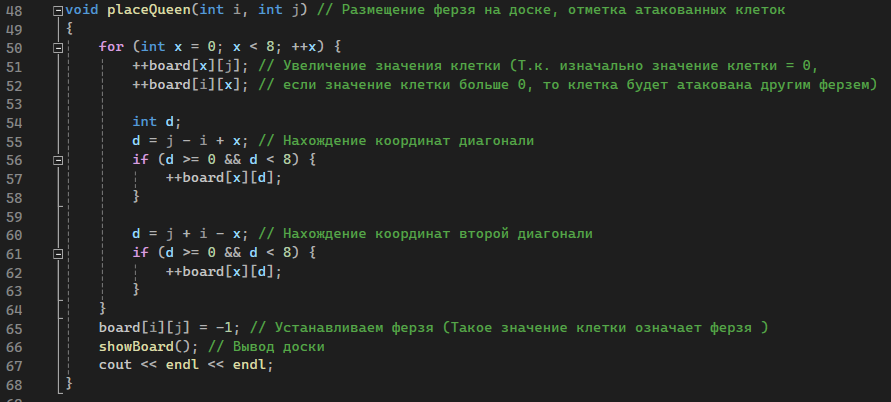
1. ***i=3, j=0, board[3][0] == 2***
2. ***i=3, j=1, board[3][1] == 2***
3. ***i=3, j=2, board[3][2] == 1***
4. ***PlaceQeen(3, 2) // result = True***
5. ***if (!(result = checkQueen(1 + 1))) break***

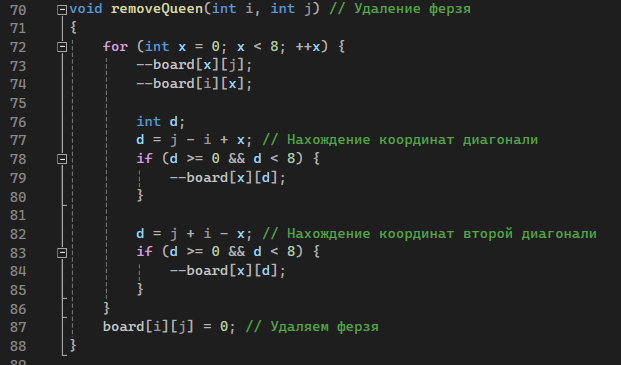
**Код для решения задачи:**

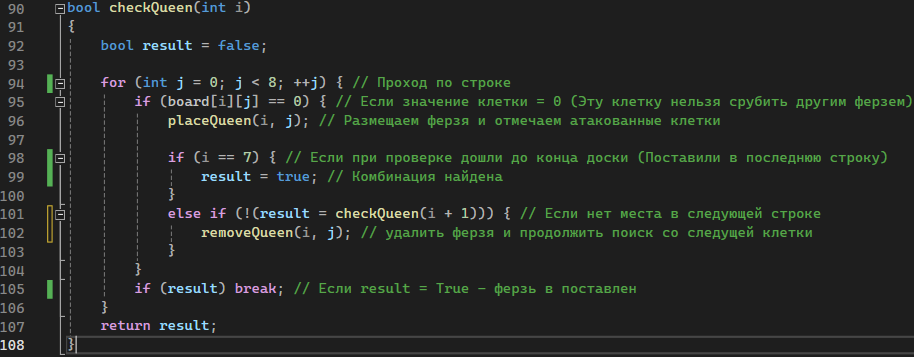
****

****

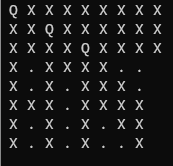
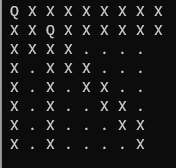
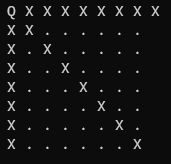
****

****

****

****

**Работа программы**

****

**…**

