

Идея решения

Решение задачи осуществляется методом перебора всех возможных вариантов расстановки ферзей, при которых они не могут срубить друг друга. Для этого условимся, что 0 – это свободная и не битая каким-либо ферзем клетка поля, 1 – клетка, в которой стоит ферзь, 2 – клетка, битая ферзем. Шахматную доску представим такой структурой данных, как двумерный массив размерностью 8x8.

В общем виде идея решения такова: необходимо поместить ферзя в первой клетке первой горизонтали (т.е. данному элементу массива присвоить значение 1), затем отметить все битые этим ферзем клетки (т.е. клетки расположенные на одной горизонтали, вертикали или диагонали с ферзем отметить в массиве двойками). Далее требуется перейти к следующей горизонтали, и в первой (с левого края горизонтали) пустой и небитой клетке поставить очередного ферзя. Для него также нужно осуществить процедуру отметки битых им полей. Таким образом, наступит один из двух вариантов развития событий: либо будет установлен восьмой ферзь в последнюю горизонталь (в таком случае, данный вариант является одним из искомым решений), либо будет установлено меньшее количество ферзей, но на поле не останется небитых клеток.

В последнем случае необходимо убрать последнего ферзя, вернуться на предыдущую горизонталь и найти следующую свободную клетку на этой горизонтали. Если таковая присутствует, то следует поставить туда ферзя и отметить битые им поля, перейти на следующую горизонталь и снова попытаться расставить ферзей в свободные клетки вплоть до последней горизонтали. Если же свободных клеток на данной горизонтали больше нет, то необходимо возвращаться на предыдущие горизонтали и убирать с них ферзей до тех пор, пока на какой либо из них не обнаружится свободная непроверенная клетка.

Процесс помещения ферзей и отметки битых полей представляет собой две рекурсивно связанные процедуры:

Процедура Ферзь

- Пока клетка не последняя в горизонтали, делать
 - Если клетка свободна - установить ферзя
 - Для этого ферзя выполнить процедуру Отметка
 - Убрать ферзя
 - Перейти к следующей клетке

Конец

Процедура Отметка

- Перебрать все клетки поля
 - Если клетка бита ферзем, отметить ее двойкой
 - Если горизонталь не последняя, перейти к следующей
 - Для нее выполнить процедуру Ферзь
 - Если кол-во ферзей – 8, вывести расстановку на экран

Конец

Особенность этих двух процедур в том, что каждая из них для своей работы требует выполнения другой процедуры. Таким образом, любая из них выполняется до того места, где вызывается вторая процедура. Это продолжается до тех пор, пока не выполнится условие завершения (в данном случае – либо достижение последней клетки горизонтали, либо последней горизонтали на доске). Тогда функционирует механизм отката – в нашем случае это означает снятие последнего ферзя с доски, позволяющее поставить нового ферзя и продолжить поиск решения.

Вариант решения на языке C

```
#include "stdafx.h"
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
void queen(int a[8][8], int m);
void output(int a[8][8]) //Вывод варианта решения
{
    for (int i=0;i<=7;i++)
    {
        for (int j=0; j<=7;j++)
            cout<<a[i][j]<<" ";
        cout<<"\n";
    }
    cout<<"\n";
}
int count(int a[8][8]) //Подсчет количества ферзей, стоящих на доске
{
    int c=0;
    for (int i=0; i<=7; i++)
    {
        for (int j=0; j<=7; j++)
            if (a[i][j] == 1) c++;
    }
    if (c==8) return 1;
    else return 0;
}
void mark(int a[8][8], int m, int n)
{
    for (int i=0; i<=7; i++) //Отметка битых полей
    {
        for (int j=0; j<=7; j++)
            if ((a[i][j]!=1) && ((abs(i-m) == abs(j-n)) || (i==m) || (j==n))) a[i][j]=2;
    }
    if (m<7) queen(a,m+1); //Переход к процедуре постановки ферзя в следующей
    //горизонтали
    if (count(a)==1) output(a); //Проверка, является ли текущая расстановка ферзей
    //решением задачи
}
void mass(int a[8][8]? int b[8][8]) //Cjplfybt rjgbb vfccbdf
{
    for (int i=0; i<=7; i++)
    {
        for (int j=0; j<=7; j++)
            b[i][j]=a[i][j];
    }
}
void queen(int a[8][8], int m)
{
```

```

int b[8][8];
mass(a,b); //Копирование рабочего массива
for (int j=0; j<=7; j++)
    if (a[m][j]==0)
        {
            a[m][j]=1; //Установка ферзя в свободное поле
            mark(a,m,j); //Переход к процедуре отметки битых полей
            mass(b,a); //Возвращение массива к исходному состоянию
        }
}

void main()
{
    int a[8][8];
    for (int i=0; i<=7; i++)
        {
            for (int j=0; j<=7; j++)
                a[i][j]=0;
        }
    i=0; //Запуск процедуры расстановки ферзей начиная с первой горизонтали
    queen(a,i);
}

```

Вариант решения на языке Компонентный Паскаль

```

MODULE Zadacha5;
IMPORT StdLog;
TYPE
    mass = ARRAY 8 OF ARRAY 8 OF INTEGER;
PROCEDURE output(a:mass); (*Вывод варианта решения*)
VAR
    i,j: INTEGER;
BEGIN
    FOR i := 0 TO 7 DO
        FOR j := 0 TO 7 DO
            StdLog.String('|');
            StdLog.Int(a[i,j]);
            StdLog.String('|');
        END;
        StdLog.String(' ');
    END;
    StdLog.String(' ');
END output;
PROCEDURE count(a: mass): INTEGER; (*Подсчет количества ферзей, стоящих на доске*)
VAR
    i,j,c: INTEGER;
BEGIN
    c:=0;
    FOR i := 0 TO 7 DO
        FOR j := 0 TO 7 DO
            IF a[i,j] = 1 THEN

```

```

                c:=c+1;
            END;
        END;
    END;
    IF c = 8 THEN RETURN 1
    ELSE RETURN 0;
    END;
END count;
PROCEDURE ^ queen(a: mass; m: INTEGER);
PROCEDURE mark(a: mass; m,n: INTEGER);
VAR
    i,j: INTEGER;
BEGIN
    FOR i := 0 TO 7 DO
        FOR j := 0 TO 7 DO
            IF ((a[i,j]#1) & ((ABS(i-m) = ABS(j-n)) OR (i=m) OR (j=n))) THEN
                a[i,j] :=2;
            END;
        END;
    END;
    IF m< 7 THEN
        queen(a,m+1);
    END;
    IF count(a) = 1 THEN
        output(a);
    END;
END mark;
PROCEDURE queen(a: mass; m: INTEGER);
VAR
    j: INTEGER;
BEGIN
    FOR j := 0 TO 7 DO
        IF a[m,j] = 0 THEN
            a[m,j]:=1; (*Установка ферзя в свободное поле*)
            mark(a,m,j); (*Переход к процедуре отметки битых полей*)
            a[m,j]:=0; (*Возвращение массива к исходному состоянию*)
        END;
    END;
END queen;
PROCEDURE main*;
VAR
    a:mass;
    i,j: INTEGER;
BEGIN
    FOR i := 0 TO 7 DO
        FOR j := 0 TO 7 DO
            a[i,j] := 0;
        END;
    END;
END;

```

```
        i:=0;  
горизонтали*)  
        queen(a,i);  
END main;  
END Zadacha5.
```

(*Запуск процедуры расстановки ферзей начиная с первой