Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение

высшего образования

«**Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные

системы»

**ОТЧЁТ**

**по лабораторной работе №17**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Рекурсивные функции. Задача о расстановке 8 ферзей

Вариант 17

Выполнил:

Студент группы ИВТ-20-2б

Сафронов Владислав Владиславович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

**Пермь, 2021**

**Цель задачи**

Научиться применять алгоритм поиска с возвратом.

**Постановка задачи**

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

**Анализ задачи**

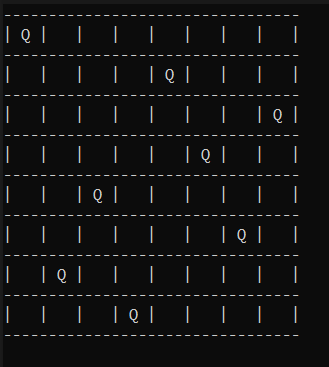
1. Шахматная доска реализована с использованием глобального динамического двумерного массива. Свободные клетки: 0, занятые клетки: -1, отличные числа – клетки под боем.
2. int board[8][8];
3. Функция setQueen заполняет клетки главной и побочной диагонали атаки ферзя, его расположение, клетки столбца и строки.
4. void SetQueen(int x, int y)
5. {
6. for (int i = 0; i < 8; ++i)
7. {
8. ++board[i][y];
9. ++board[x][i];
10. int diag = y - x + i;
11. if (diag >= 0 && diag < 8) ++board[i][diag];
12. diag = y + x - i;
13. if (diag >= 0 && diag < 8) ++board[i][diag];
14. board[x][y] = -1;
15. }
16. }
17. Функция reset удаляет ферзя из поля.
18. void Reset(int x, int y)
19. {
20. for (int i = 0; i < 8; ++i)
21. {
22. --board[i][y];
23. --board[x][i];
24. int diag = y - x + i;
25. if (diag >= 0 && diag < 8) --board[i][diag];
26. diag = y + x - i;
27. if (diag >= 0 && diag < 8) --board[i][diag];
28. board[x][y] = 0;
29. }
30. }
31. Функция tryQueen проверяет строку на возможность поставить ферзя. Функия вызывается рекурсивно. В цикле перебираются клетки поля, если найдена клетка со значением 0, то в эту клетку ставится ферзь с помощью функции setQueen.
32. bool TryQueen(int i)
33. {
34. bool res = false;
35. for (int j = 0; j < 8 && !res; ++j)
36. {
37. if (board[i][j] == 0)
38. {
39. SetQueen(i, j);
40. if (i == 7) res = !res;
41. else if (!(res = TryQueen(i + 1))) Reset(i, j);
42. }
43. }
44. return res;
45. }

**Блок-схема программы**

**Код программы**

1. #include <iostream>
2. using namespace std;
4. int board[8][8];
6. void ShowBoard()
7. {
8. cout << "---------------------------------\n";
9. for (int i = 0; i < 8; ++i)
10. {
11. for (int j = 0; j < 8; ++j)
12. {
13. if (board[i][j] == -1) cout << "| Q ";
14. else cout << "| ";
15. }
16. cout << "|\n---------------------------------\n";
17. }
18. }
20. void SetQueen(int x, int y)
21. {
22. for (int i = 0; i < 8; ++i)
23. {
24. ++board[i][y];
25. ++board[x][i];
26. int diag = y - x + i;
27. if (diag >= 0 && diag < 8) ++board[i][diag];
28. diag = y + x - i;
29. if (diag >= 0 && diag < 8) ++board[i][diag];
30. board[x][y] = -1;
31. }
32. }
34. void Reset(int x, int y)
35. {
36. for (int i = 0; i < 8; ++i)
37. {
38. --board[i][y];
39. --board[x][i];
40. int diag = y - x + i;
41. if (diag >= 0 && diag < 8) --board[i][diag];
42. diag = y + x - i;
43. if (diag >= 0 && diag < 8) --board[i][diag];
44. board[x][y] = 0;
45. }
46. }
48. bool TryQueen(int i)
49. {
50. bool res = false;
51. for (int j = 0; j < 8 && !res; ++j)
52. {
53. if (board[i][j] == 0)
54. {
55. SetQueen(i, j);
56. if (i == 7) res = !res;
57. else if (!(res = TryQueen(i + 1))) Reset(i, j);
58. }
59. }
60. return res;
61. }
63. int main()
64. {
65. for (int i = 0; i < 8; ++i)
66. for (int j = 0; j < 8; ++j) board[i][j] = 0;
67. TryQueen(0);
68. ShowBoard();
69. }

**Скриншоты результатов работы программы**

****