

Задача №4

Загадка с n ферзями — это задача о том, как разместить n ферзей на $n \times n$ шахматной доске так, чтобы никакие два ферзя не атаковали друг друга.

Учитывая целое число n , верните все различные решения головоломки с n ферзями.

Вы можете вернуть ответ в любом порядке.

Каждое решение содержит отдельную конфигурацию доски для размещения n ферзей, где 'Q' и '.' оба обозначают ферзя и пустое место соответственно.

Пример 1:

Вход: $n = 4$

Выход: `[[".Q..", "...Q", "Q...", "..Q."], [".Q.", "Q..", "...Q", ".Q.."]]`

Пояснение: Существует два различных решения головоломки с четырьмя ферзями, как показано выше.

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

class Solution {
public:
    vector<vector<string>> solveNQueens(int n) {
        vector<vector<string>> results;
        vector<string> board(n, string(n, '.'));
        vector<int> queens(n, -1);
        backtrack(results, board, queens, 0, n);
        return results;
    }

private:
    void backtrack(vector<vector<string>>& results, vector<string>& board, vector<int>& queens, int row, int n) {
        if (row == n) {
            results.push_back(board);
            return;
        }

        for (int col = 0; col < n; col++) {
            if (isValid(board, queens, row, col, n)) {
                board[row][col] = 'Q';
                queens[row] = col;
                backtrack(results, board, queens, row + 1, n);
                board[row][col] = '.';
                queens[row] = -1;
            }
        }
    }

    bool isValid(vector<string>& board, vector<int>& queens, int row, int col, int n) {
        for (int i = 0; i < row; i++) {
            if (board[i][col] == 'Q') {
                return false;
            }
        }

        for (int i = row - 1, j = col - 1; i >= 0 && j >= 0; i--, j--) {
            if (board[i][j] == 'Q') {
                return false;
            }
        }

        for (int i = row - 1, j = col + 1; i >= 0 && j < n; i--, j++) {
            if (board[i][j] == 'Q') {
                return false;
            }
        }

        return true;
    }
};

int main() {
    int n = 4;
    Solution solution;
    vector<vector<string>> results = solution.solveNQueens(n);

    cout << "Solving puzzles with " << n << " queens:" << endl;
    for (const vector<string>& result : results) {
        for (const string& row : result) {
            cout << row << endl;
        }
        cout << endl;
    }

    return 0;
}
```



Solving puzzles with 4 queens:



.Q..



...Q



Q...

..Q.

..Q.

Q...

...Q

.Q..