The N-Queens Problem and C++

By Caden Roberts

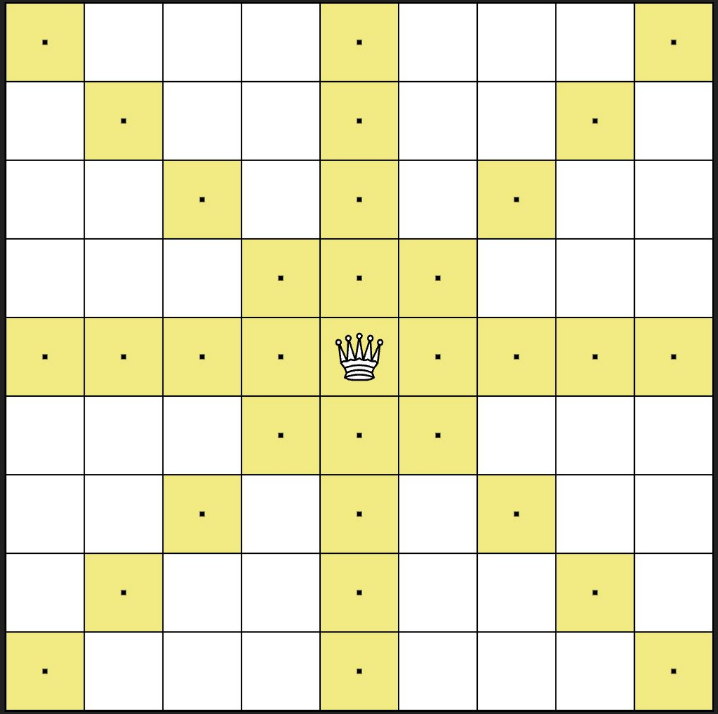
What is N-Queens?

Computer Science problem:

1. Using Queens from the game of chess

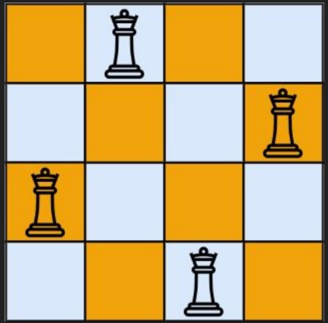
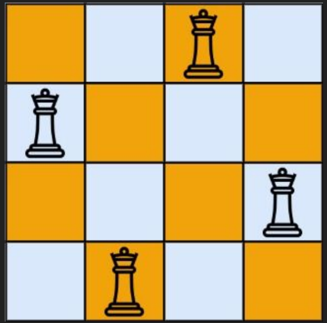
2. For a chessboard of size N by N

3. How can N Queens be placed such that no queens threaten each other



History of N-Queens

- In 1848, the *8*-Queens problem was posed by chess composer Max Bezzel
- In 1850, Franz Nauck published the first solutions and extended the problem to the N-Queens problem

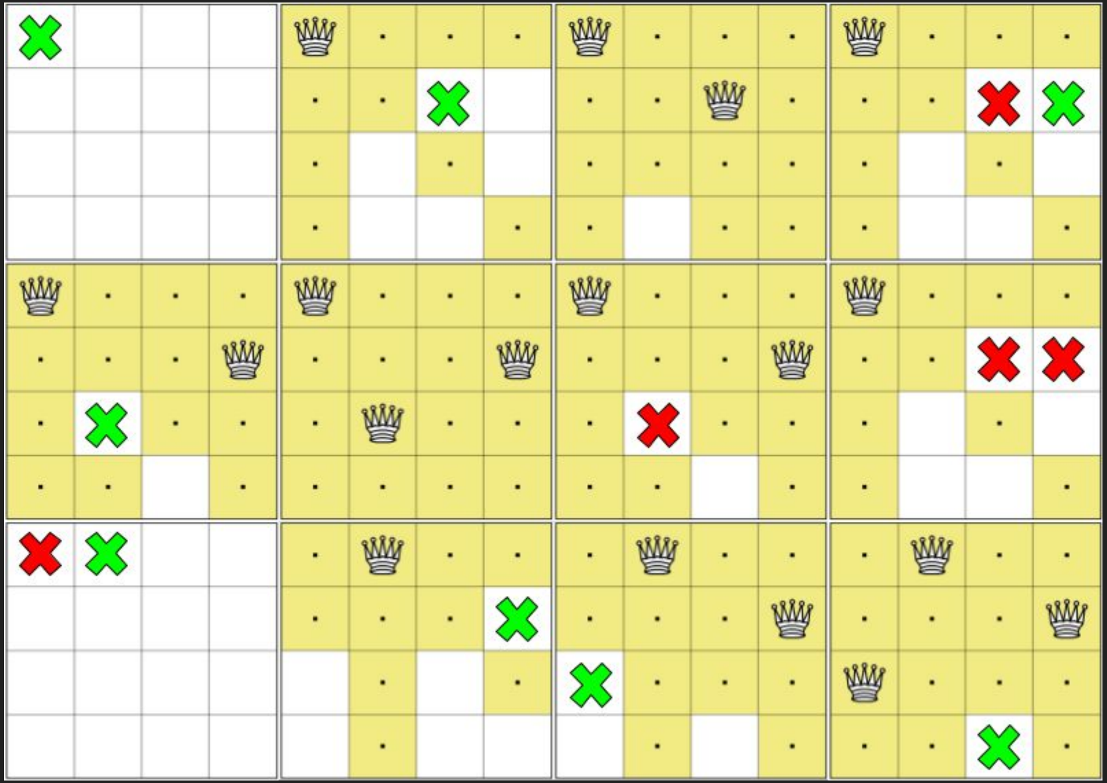



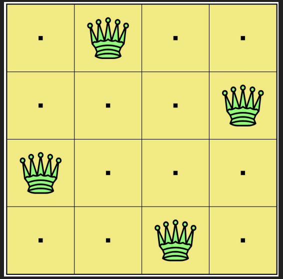
Common N-Queens Challenges

- Find *a* solution to N-Queens
  - Multiple approaches
- Find *all* solutions to N-Queens
  - Runtime, # of solutions is only know up to N=27
- Find *a/all* solution(s) to N-Queens, with any amount of preset Queens
  - Possible there is no solution

What is backtracking?

- If a Queen is in conflict when placed, it must move right to the next column
- If there is no column to go to, we must place it off the board
- Now, ‘backtrack’ and try to place the previous queen in its next column until it is either out of conflict or off the board
- If the queen is safe we can move back on to the next row, otherwise, ‘backtrack’ again





Backtracking in N Queens Example

1. Go row by row, starting with the leftmost column
2. Place the queen or Move over 1 column if it was backtracked to
3. If the queen is in conflict in its column and not off the board, move it to the next column until isn't
4. If the queen is placed off the board, backtrack to step 2
5. Either all queens will be placed or the 1st row queen will be forced to be placed off of the board

Pseudo-ish Code for Coding Backtracking N Queens

```
Array “ of N ints                                      )  Variables
2 integers ‘row’ and ‘column’ equal 0                  )
While row is less than N :  )  Until Solution
    Set solution[row] to column                        )
    If there is conflict:                              )  Place Queen,
        Increment column                               )  Check Conflict,
    Else:                                              )  Adjust Variables
        Increment row                                  )
        Set column to 0                                )
    While column equals N and row is greater than 0:           )  Backtrack,
        Decrement row, set column to the value at that row + 1 )  Adjust variables
    If row equals 0 and column equals N:   )  Check No solution
        No solution                        )
Solution  )  Solution
```

My C++ code to find a solution to N queens

```cpp
#include <iostream>
using namespace std;
int main () {
    int n = -1, column = 0, row = 0, conflict = 0;
    while (n < 0) {
        cout << "Number of Queens? (n >= 0): ";
        cin >> n;
    }
    int solution[n];
    while (row < n && ! ( row==0 && column==n )) {
        solution[row] = column;
        for (int next = 0; next < row; next++) {
            if (solution[next]==column || abs(solution[next] - column) == abs(next - row)) {
                conflict = 1;
                break;
            }
        }
        if (conflict) {
            column++;
            conflict = 0;
        }
        else {
            row++;
            column = 0;
        }
        while (column == n && row > 0) column = solution[--row] + 1;
    }
    if (row==0 && column==n) cout << "No solution.\n";
    else {
        for (int topborder = 0; topborder < n*2 + 3; topborder++) cout << "-";
        cout << "\n";
        for (int board = 0; board < n; board++) {
            cout << "| ";
            for (int queen = 0; queen<n; queen++) {
                if (queen==solution[board]) cout << "Q ";
                else cout << ". ";
            }
            cout << "|\n";
        }
        for (int bottomborder = 0; bottomborder < n*2 + 3; bottomborder++) cout << "-";
    }
    return 0;
}
```

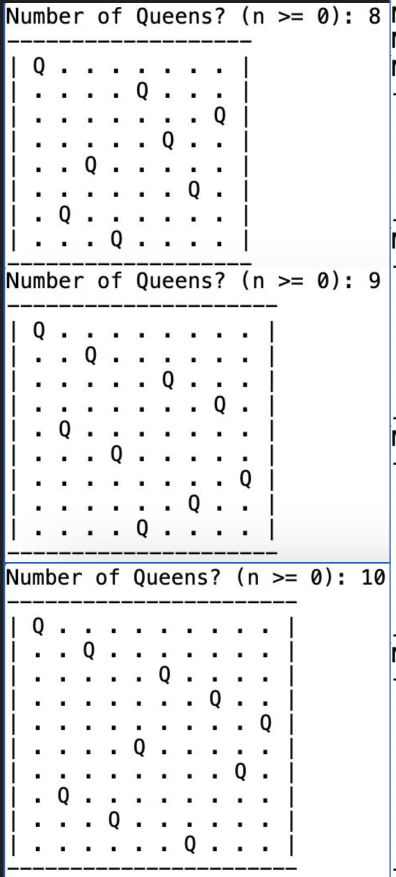
Variables

Until Solution or No Solution

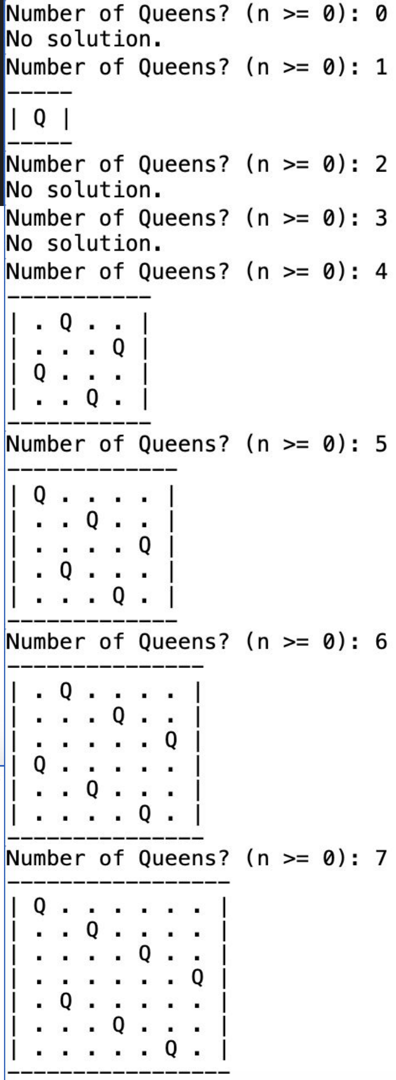
Place Queens, Check Conflict

Backtrack

Print Solution/ None



```
Number of Queens? (n >= 0): 8
-------------------
| Q . . . . . . . |
| . . . . Q . . . |
| . . . . . . . Q |
| . . . . . Q . . |
| . . Q . . . . . |
| . . . . . . Q . |
| . Q . . . . . . |
| . . . Q . . . . |
-------------------
Number of Queens? (n >= 0): 9
---------------------
| Q . . . . . . . . |
| . . Q . . . . . . |
| . . . . . Q . . . |
| . . . . . . . Q . |
| . Q . . . . . . . |
| . . . Q . . . . . |
| . . . . . . . . Q |
| . . . . . . Q . . |
| . . . . Q . . . . |
---------------------
Number of Queens? (n >= 0): 10
-----------------------
| Q . . . . . . . . . |
| . . Q . . . . . . . |
| . . . . . Q . . . . |
| . . . . . . . Q . . |
| . . . . . . . . . Q |
| . . . . Q . . . . . |
| . . . . . . . . Q . |
| . Q . . . . . . . . |
| . . . Q . . . . . . |
| . . . . . . Q . . . |
-----------------------
```



```
Number of Queens? (n >= 0): 0
No solution.
Number of Queens? (n >= 0): 1
-----
| Q |
-----
Number of Queens? (n >= 0): 2
No solution.
Number of Queens? (n >= 0): 3
No solution.
Number of Queens? (n >= 0): 4
-----------
| . Q . . |
| . . . Q |
| Q . . . |
| . . Q . |
-----------
Number of Queens? (n >= 0): 5
-------------
| Q . . . . |
| . . Q . . |
| . . . . Q |
| . Q . . . |
| . . . Q . |
-------------
Number of Queens? (n >= 0): 6
---------------
| . Q . . . . |
| . . . Q . . |
| . . . . . Q |
| Q . . . . . |
| . . Q . . . |
| . . . . Q . |
---------------
Number of Queens? (n >= 0): 7
-----------------
| Q . . . . . . |
| . . Q . . . . |
| . . . . Q . . |
| . . . . . . Q |
| . Q . . . . . |
| . . . Q . . . |
| . . . . . Q . |
-----------------
```

My C++ code to find all solutions to N queens

```cpp
#include <iostream>
#include <vector>
using namespace std;
int main () {
    int n = 0, column = 0, row = 0, conflict = 0, boardcount = 0, solutioncount = 0;
    while (n <= 0) {
        cout << "Number of Queens? (n>0) : ";
        cin >> n;
    }
    int solution[n];
    vector<int> solutions;
    while (!(row==0 && column==n)) {
        while (row < n && !(row==0 && column==n)) {
            solution[row] = column;
            conflict = 0;
            for (int i = 0; i<row; i++) {
                if (solution[i]==column || abs(solution[i] - column) == abs(i - row)) {
                    conflict = 1;
                    break;
                }
            }
            if (conflict) column++;
            else {
                row++;
                column = 0;
            }
            while (column==n && row>0) column = solution[--row]+1;
        }
        if (!(row==0 && column==n)) {
            for (int a = 0; a<n; a++) solutions.push_back(solution[a]);
            solutioncount++;
            row--;
            column = solution[row]+1;
            while (column==n && row>0)  column = solution[--row]+1;
        }
    }
    for (int a : solutions) {
        if (boardcount % n == 0) {
            for (int i = 0; i < n*2 + 3; i++) cout << "-";
            cout << "\n";
        }
        cout << "| ";
        for (int i = 0; i<n; i++) {
            if (i==a) cout << "Q ";
            else cout << ". ";
        }
        cout << "|\n";
        if (boardcount++ % n == n-1) {
            for (int i = 0; i < n*2 + 3; i++) cout << "-";
            cout << "\n";
        }
    }
    cout << "There " << (solutioncount == 1 ? "is " : "are ") << solutioncount << " total solution" <<
( solutioncount == 1 ? ".\n" : "s.\n");
    return 0;
}
```

Variables

Until Every Possibility

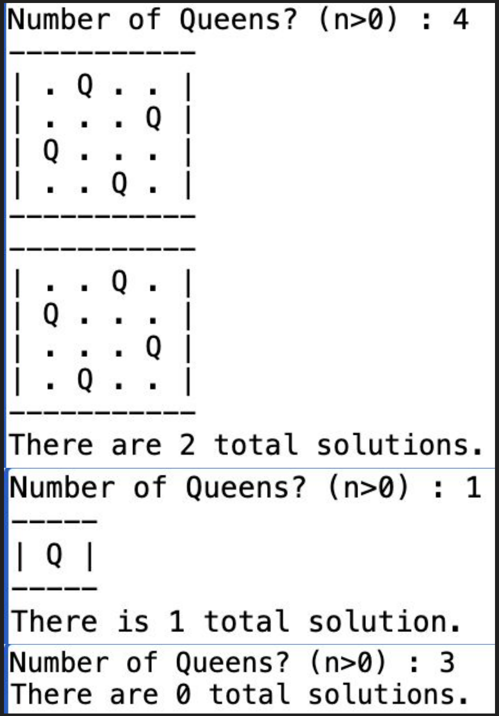
Until Solution or No Solution

Place Queens,
Check Conflict

Backtrack

Add Solution,
Continue/Backtrack

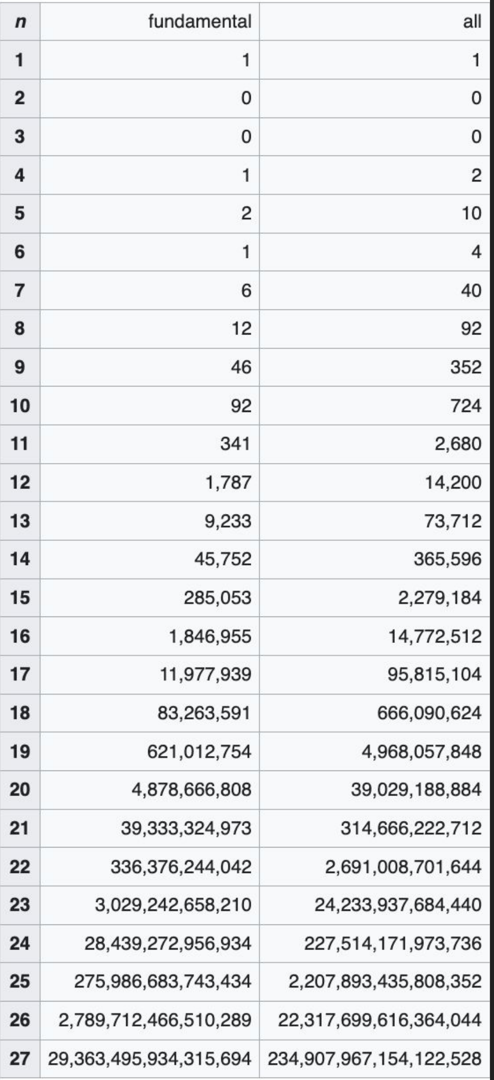
Print Solutions



```
Number of Queens? (n>0) : 4
-----------
| . Q . . |
| . . . Q |
| Q . . . |
| . . Q . |
-----------
-----------
| . . Q . |
| Q . . . |
| . . . Q |
| . Q . . |
-----------
There are 2 total solutions.
```

```
Number of Queens? (n>0) : 1
-----
| Q |
-----
There is 1 total solution.
```

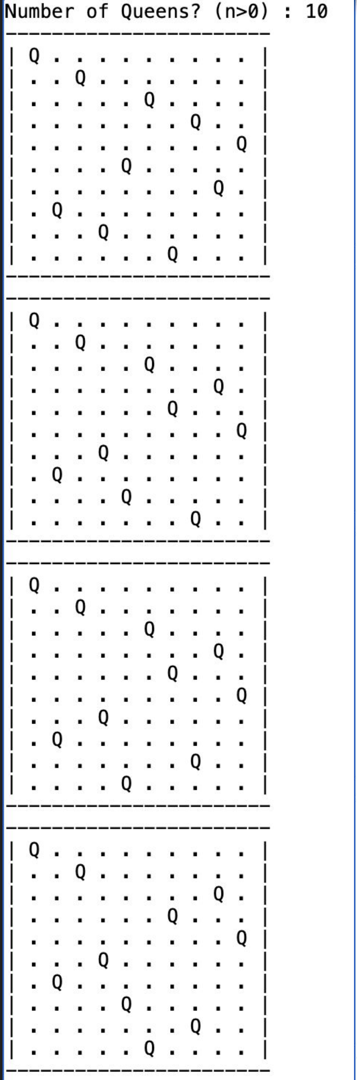
```
Number of Queens? (n>0) : 3
There are 0 total solutions.
```



n	fundamental	all
1	1	1
2	0	0
3	0	0
4	1	2
5	2	10
6	1	4
7	6	40
8	12	92
9	46	352
10	92	724
11	341	2,680
12	1,787	14,200
13	9,233	73,712
14	45,752	365,596
15	285,053	2,279,184
16	1,846,955	14,772,512
17	11,977,939	95,815,104
18	83,263,591	666,090,624
19	621,012,754	4,968,057,848
20	4,878,666,808	39,029,188,884
21	39,333,324,973	314,666,222,712
22	336,376,244,042	2,691,008,701,644
23	3,029,242,658,210	24,233,937,684,440
24	28,439,272,956,934	227,514,171,973,736
25	275,986,683,743,434	2,207,893,435,808,352
26	2,789,712,466,510,289	22,317,699,616,364,044
27	29,363,495,934,315,694	234,907,967,154,122,528

Fundamental Solutions

- For 8 queens, there are 92 solutions
- For solutions that differ by operations of rotation and reflection, there are only 12
- The first and last solutions are symmetric (second and second to last, etc.)
- This also shows how backtracking works, finding the very first and then the very last possible solution
- It would take extremely long to print all solutions of large N's

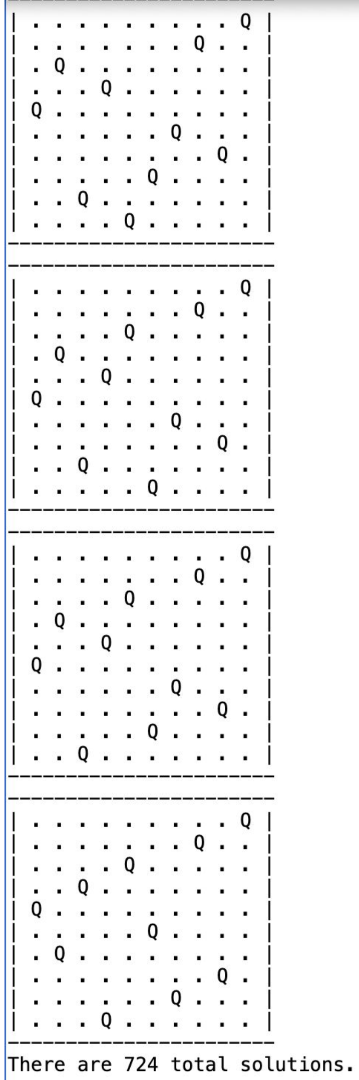


```
Number of Queens? (n>0) : 10
-----------------------
| Q . . . . . . . . . |
| . . Q . . . . . . . |
| . . . . . Q . . . . |
| . . . . . . . Q . . |
| . . . . . . . . . Q |
| . . . . Q . . . . . |
| . . . . . . . . Q . |
| . Q . . . . . . . . |
| . . . Q . . . . . . |
| . . . . . . Q . . . |
-----------------------

-----------------------
| Q . . . . . . . . . |
| . . Q . . . . . . . |
| . . . . . Q . . . . |
| . . . . . . . . Q . |
| . . . . . . Q . . . |
| . . . . . . . . . Q |
| . . . Q . . . . . . |
| . Q . . . . . . . . |
| . . . . Q . . . . . |
| . . . . . . . Q . . |
-----------------------

-----------------------
| Q . . . . . . . . . |
| . . Q . . . . . . . |
| . . . . . Q . . . . |
| . . . . . . . . Q . |
| . . . . . . Q . . . |
| . . . . . . . . . Q |
| . . . Q . . . . . . |
| . Q . . . . . . . . |
| . . . . . . . Q . . |
| . . . . Q . . . . . |
-----------------------

-----------------------
| Q . . . . . . . . . |
| . . Q . . . . . . . |
| . . . . . . . . Q . |
| . . . . . . Q . . . |
| . . . . . . . . . Q |
| . . . Q . . . . . . |
| . Q . . . . . . . . |
| . . . . Q . . . . . |
| . . . . . . . Q . . |
| . . . . . Q . . . . |
-----------------------
```



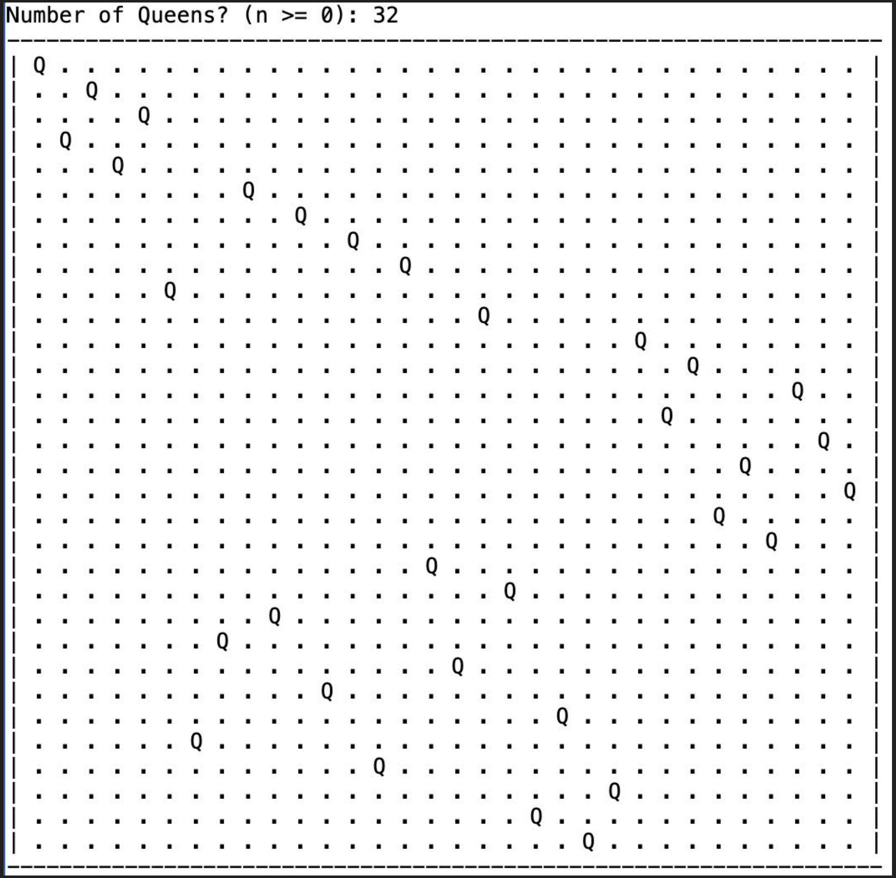
```
-----------------------
| . . . . . . . . . Q |
| . . . . . . . Q . . |
| . Q . . . . . . . . |
| . . . Q . . . . . . |
| Q . . . . . . . . . |
| . . . . . . Q . . . |
| . . . . . . . . Q . |
| . . . . . Q . . . . |
| . . Q . . . . . . . |
| . . . . Q . . . . . |
-----------------------

-----------------------
| . . . . . . . . . Q |
| . . . . . . . Q . . |
| . . . . Q . . . . . |
| . Q . . . . . . . . |
| . . . Q . . . . . . |
| Q . . . . . . . . . |
| . . . . . . Q . . . |
| . . . . . . . . Q . |
| . . Q . . . . . . . |
| . . . . . Q . . . . |
-----------------------

-----------------------
| . . . . . . . . . Q |
| . . . . . . . Q . . |
| . . . . Q . . . . . |
| . Q . . . . . . . . |
| . . . Q . . . . . . |
| Q . . . . . . . . . |
| . . . . . . Q . . . |
| . . . . . . . . Q . |
| . . . . . Q . . . . |
| . . Q . . . . . . . |
-----------------------

-----------------------
| . . . . . . . . . Q |
| . . . . . . . Q . . |
| . . . . Q . . . . . |
| . . Q . . . . . . . |
| Q . . . . . . . . . |
| . . . . . Q . . . . |
| . Q . . . . . . . . |
| . . . . . . . . Q . |
| . . . . . . Q . . . |
| . . . Q . . . . . . |
-----------------------
There are 724 total solutions.
```

Takes about 35 seconds to complete.



```
Number of Queens? (n >= 0): 32
-------------------------------------------------------------------
| Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . |
| . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . |
| . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q |
| . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . |
| . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . |
| . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . |
| . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . |
| . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . |
-------------------------------------------------------------------
```

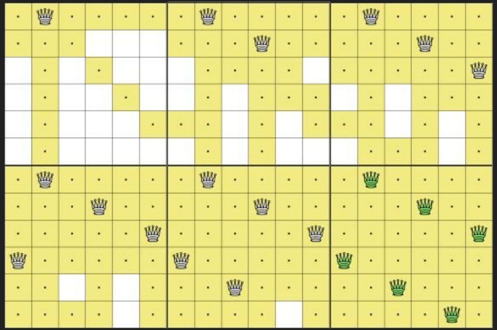
If we only need 1 solution...

If the remainder from dividing n by 6 is not 2 or 3:

1. Place queens at all even numbered columns ascending from the 2nd column
2. For the remaining odd numbered columns place queens ascending from the 1st column

1

2

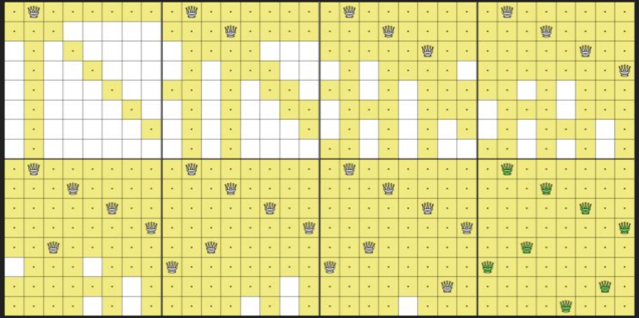


If the remainder is 2:

1. Place queens at all even numbered columns ascending from the 2nd column
2. For the remaining odd columns place queens descending from the 3rd column, then ascending from the 7th column, and finally place a queen in the 5th column

1

2

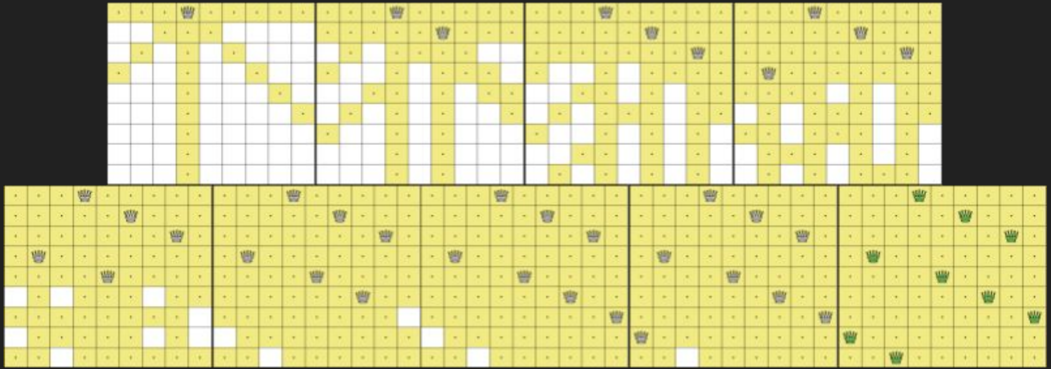


If the remainder is 3:

1. Place queens at all even numbered columns ascending from the 4th and then place a queen in the 2nd column
2. For the remaining odd numbered columns place queens ascending from the 5th column and then ascending from the 1st column

1

2



C++ code to get a solution to N queens

```cpp
#include <iostream>
#include <vector>
using namespace std;
int main () {
    int n = 0, b = 0;
    while (n <= 0) {
        cout << "Number of Queens? (n>0) : ";
        cin >> n;
    }
    vector<int> solutions;
    if (n == 2 || n == 3) {
        cout << "No solution.\n";
        return 0;
    }  else if (n % 6 == 3) {
        for (int i = 4; i <= n; i+=2) solutions.push_back(i);
        solutions.push_back(2);
        for (int i = 5; i <=n; i+=2) solutions.push_back(i);
        solutions.push_back(1);
        solutions.push_back(3);
    } else {
        for (int i = 2; i <= n; i+=2) solutions.push_back(i);
        if (n % 6 == 2) {
            solutions.push_back(3);
            solutions.push_back(1);
            for (int i = 7; i <= n; i+=2) solutions.push_back(i);
            solutions.push_back(5);
        } else {
            for (int i = 1; i <= n; i+=2) solutions.push_back(i);
        }
    }
    for (int a : solutions) {
        if (b % n == 0) {
            for (int i = 0; i < n*2 + 3; i++) cout << "-";
            cout << "\n";
        }
        cout << "| ";
        for (int i = 1; i<=n; i++) {
            if (i==a) cout << "Q ";
            else cout << ". ";
        }
        cout << "|\n";
        if (b++ % n == n-1) {
            for (int i = 0; i < n*2 + 3; i++) cout << "-";
            cout << "\n";
        }
    }
    return 0;
}
```

Variables

N = 2 or 3

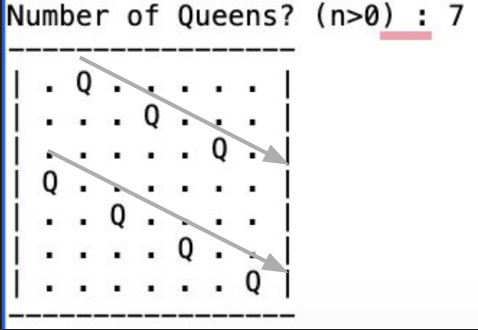
N % 6 = 3

N % 6 = 2

Otherwise

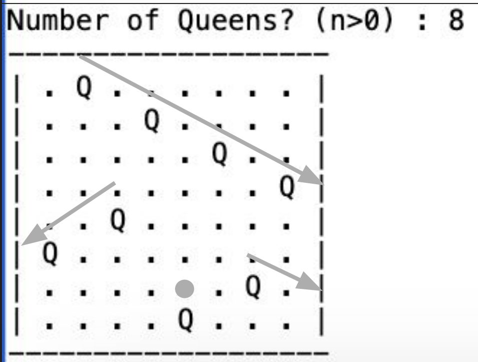
Print Solution

7 % 6 = 1



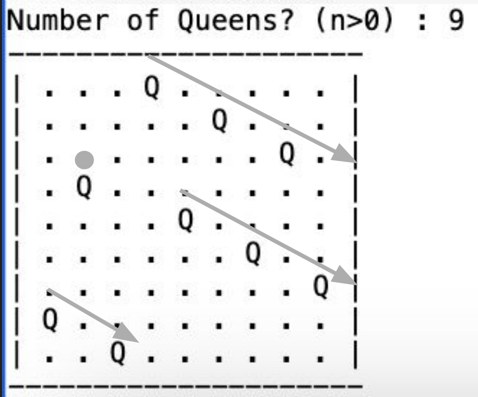
```
Number of Queens? (n>0) : 7
-----------------
| . Q . . . . . |
| . . . Q . . . |
| . . . . . Q . |
| Q . . . . . . |
| . . Q . . . . |
| . . . . Q . . |
| . . . . . . Q |
-----------------
```

8 % 6 = 2



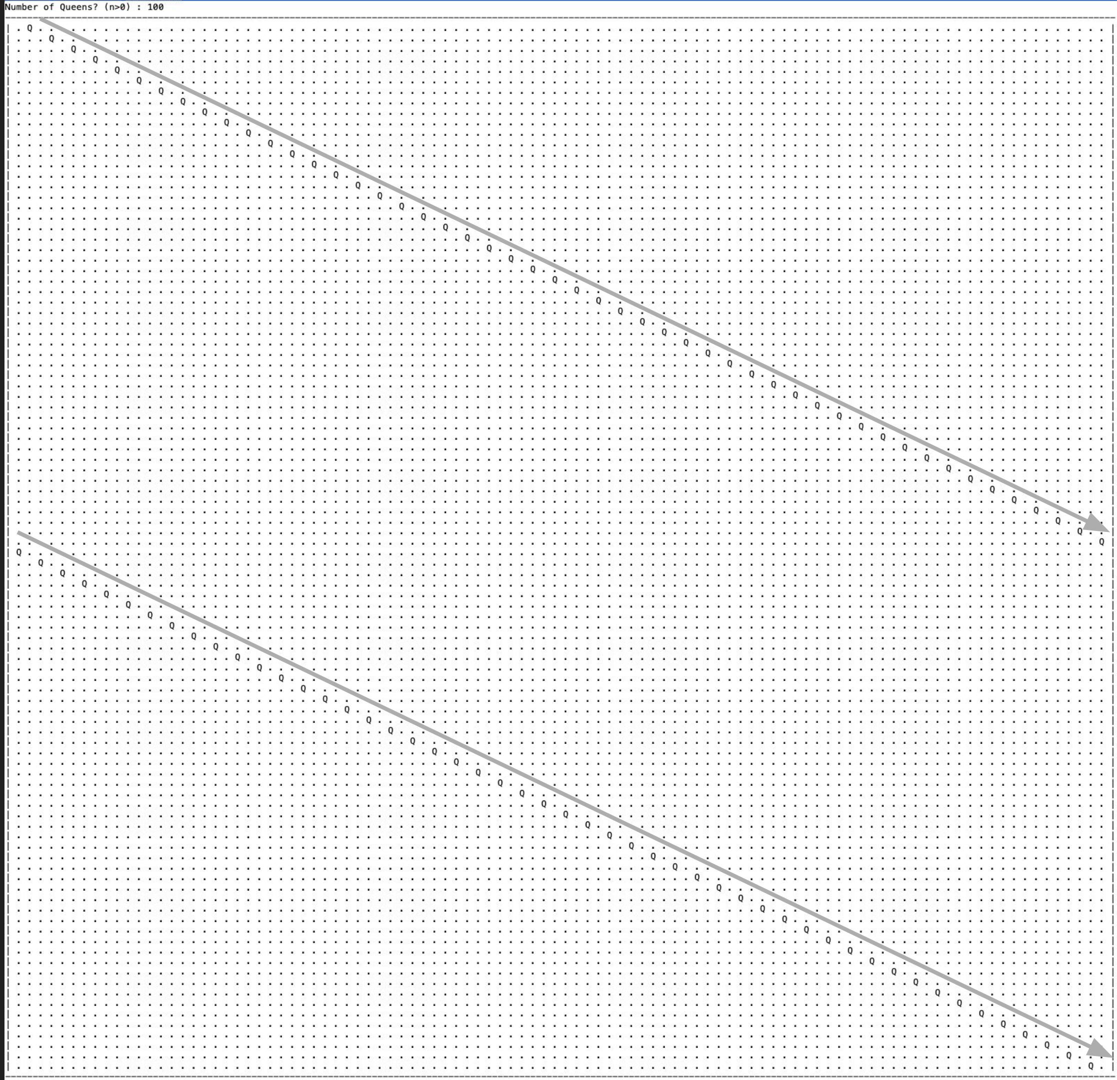
```
Number of Queens? (n>0) : 8
-------------------
| . Q . . . . . . |
| . . . Q . . . . |
| . . . . . Q . . |
| . . . . . . . Q |
| . . Q . . . . . |
| Q . . . . . . . |
| . . . . . . Q . |
| . . . . Q . . . |
-------------------
```

9 % 6 = 3



```
Number of Queens? (n>0) : 9
---------------------
| . . . Q . . . . . |
| . . . . . Q . . . |
| . . . . . . . Q . |
| . Q . . . . . . . |
| . . . . Q . . . . |
| . . . . . . Q . . |
| . . . . . . . . Q |
| Q . . . . . . . . |
| . . Q . . . . . . |
---------------------
```

N = 100 is instant now.
N%6 = 4



```
Number of Queens? (n>0) : 100
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . . . |
| . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Q . |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Thank you!
For fun:

linkedin.com/games/queens

References

[1] “4 Queens Puzzle,” *Brainbashers.com*, 2024. https://www.brainbashers.com/queens.asp?size=4 (accessed Nov. 15, 2024).

[2] *https://ih1.redbubble.net/image.47109792.4236/flat,750x1000,075,t.u4.jpg.*

[3] Wikipedia Contributors, “C++,” *Wikipedia*, Mar. 25, 2019. https://en.wikipedia.org/wiki/C%2B%2B

[4] *https://www.puzzleprime.com/puzzles/brain-teasers/deduction/eight-queens-puzzle/.*

[5] *Geeksforgeeks.org*, 2024. https://media.geeksforgeeks.org/wp-content/uploads/20230814111826/Backtracking.png (accessed Nov. 15, 2024).

[6] “Eight queens puzzle,” *Wikipedia*, Apr. 22, 2021. https://en.wikipedia.org/wiki/Eight_queens_puzzle

How do we find a solution?
- Algorithms of N-Queens

- Backtracking
- Brute-Force
- Min-Conflicts
- Unique Methods
- Etc.