Code:

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

int count = 0;

void print1D(int board[], int n) {
  for (int i = 0; i < n; i++) {
    cout << board[i] << " ";
  }
  cout << endl;
}

void print2D(int board[], int n) {
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
      if (board[i] == j) {
        cout << "Q ";
      } else {
        cout << ". ";
      }
    }
    cout << endl;
  }
  cout << endl;
}

bool isSafe(int board[], int row, int col, int n) {

  for (int i = 0; i < row; i++) {
    if (board[i] == col) {
      return false;
    }
  }

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

  return true;
```

```
}

void solveNQueen(int board[], int row, int n) {

  if (row == n) {
    count++;
    cout<<endl;
    print1D(board, n);
    cout<<endl;
    print2D(board, n);
    return;
  }

  for (int col = 0; col < n; col++) {
    if (isSafe(board, row, col, n)) {
      board[row] = col;
      solveNQueen(board, row + 1, n);

      board[row] = -1;
    }
  }
}

int main() {
  int n ;
  cout<<"\nEnter the value of n: ";
  cin>>n;

  int board[n];
  for (int i = 0; i < n; i++) {
    board[i] = -1;
  }
  solveNQueen(board, 0, n);
  cout << "Total number of calls to queen procedure: " << count << endl;
  return 0;
}
```

Sample Output:

```
Enter the value of n: 4
1 3 0 2

. Q . .
. . . Q
Q . . .
. . Q .
```

2 0 3 1

. . Q .
Q . . .
. . . Q
. Q . .

Total number of calls to queen procedure: 2