

Assignment No. 8

Aim: Implement C++ program for solving N-Queen's problem using Backtracking.

Theory:

- **What is Backtracking?**

In backtracking, we start with one possible move out of many available moves. We then try to solve the problem. If we are able to solve the problem with the selected move then we will print the solution. Else we will back-track and select some other move and try to solve it. If none of the moves works out we claim that there is no solution for the problem.

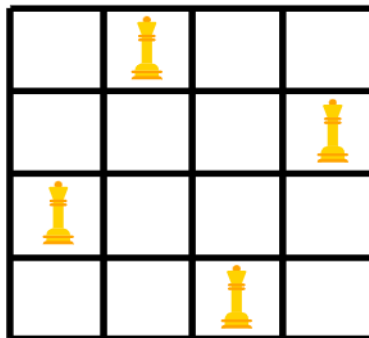
- **What is N-Queen Problem?**

This problem is commonly seen for $N=4$ and $N=8$.

Let's look at an example where $N=4$

Before solving the problem, you need to know about the movement of the queen in chess. A queen can move any number of steps in any direction. The only constraint is that it can't change its direction while it's moving. One thing that is clear by looking at the queen's movement is that no two queens can be in the same row or column. That allows us to place only one queen in each row and each column.

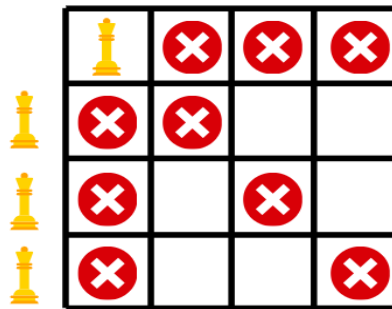
When $N=4$, the solution looks like:



Solution to the N-Queen Problem:

The way we try to solve this is by placing a queen at a position and trying to rule out the possibility of it being under attack. We place one queen in each row/column. If we see that the queen is under attack at its chosen position, we try the next position. If a queen is under attack at all the positions in a row, we backtrack and change the position of the queen placed prior to the current position. We repeat this process of placing a queen and backtracking until all the N queens are placed successfully.

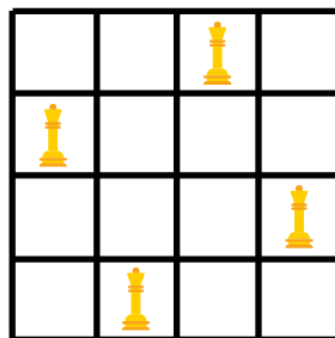
The step by step backtracking is shown as follows:



Start

The red cross marks the positions which are under attack from a queen. Whenever we reach a state where we have a queen to place but all the positions in the rows are under attack, we backtrack.

This is not the only possible solution to the problem. If you move each queen one step forward in a clockwise manner, you get another solution.



In this example we placed the queens according to rows, we can do the same thing column-wise also. In that case, each queen will belong to a column.

Program:

Output:

Conclusion: