Experiment No. 07

I. Aim: Implement N-Queen's problem using Back Tracking.

II. Theory:

The problem of N-Queens is to place N queens on an n × n chess board so that no two queens attack each other by being in the same row or same column or same diagonal.
Example: Placing 4 Queens on a 4 × 4 chess board.

Write stepwise procedure to place 5 queens on 5 × 5 chess board so that no two queens attack each other. Show atleast three possible placements with suitable diagrams.

_____
_____
_____
_____

III. Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int board [20];
int count = 0;

bool is_safe(int row, int col)
{
   for (int i = 0; i < row; i++)
   {
      if (board[i] == col || abs(board[i] - col) == abs (i - row))
      {
         return false;
      }
   }
   return true;
}

void print_solution(int n)
{
   count++;
   printf("\nSolution #%d:\n", count);
   for (int i = 0; i < n; i++)
   {
      for (int j = 0; j < n; j++)
      {
         if (board[i] == j)
         {
            printf("Q\t");
         }
         else
         {
            printf("*\t");
         }
```

```
        }
        printf("\n");
    }
}

void solve_n_queens(int row, int n)
{
    if (row == n)
    {
        print_solution(n);
        return;
    }

    for (int col = 0; col < n; col++)
    {
        if (is_safe(row, col))
        {
            board[row] = col;
            solve_n_queens(row + 1, n);
        }
    }
}

int main() {
    int n;
    printf("Enter the number of Queens (N): ");
    scanf("%d", &n);

    if (n <= 0)
    {
        printf("Number of Queens must be a positive integer.\n");
        return 1;
    }

    solve_n_queens(0, n); // Start solving from the first row (row 0)

    if (count == 0)
    {
        printf("\nNo solutions found for N = %d\n", n);
    }
    else
    {
        printf("\nTotal solutions found: %d\n", count);
    }
    getch();
}
```

IV. Output:

_____

_____

_____

_____

V. Conclusion: Successfully implemented a N-Queen Backtracking algorithm to find all possible placements.