N_QUEENS.c

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include<ctype.h>
int board[20],count;
int counter(int row,int n,int *c);
int place(int row,int column);
void print(int n);
void queen(int row,int n);
int counter(int row, int n,int *c)
{
    int column;
    for (column = 1; column <= n; ++column)
    {
        if (place(row, column))
        {
            board[row] = column;
            if (row == n)
               (*c)+=1;//print(n);
            else
                counter(row + 1, n,c);
        }
    }
}
int place(int row, int column)
{
    int i;
    for (i = 1; i <= row - 1; ++i)
    {
        if (board[i] == column)
            return 0;
        else if (fabs(board[i] - column) == fabs(i - row))
            return 0;
    }

    return 1;

}
void queen(int row, int n)
{
    int column;
    for (column = 1; column <= n; ++column)
    {
        if (place(row, column))
        {
            board[row] = column;
            if (row == n)
               print(n);
            else
                queen(row + 1, n);
        }
    }
}
void print(int n)
{
    int i, j;
```

```c
        printf("\n\nSolution %d:\n\n", ++count);

        for (i = 1; i <= n; ++i)
            printf("\t%d", i);

        for (i = 1; i <= n; ++i)
        {
            printf("\n\n%d", i);
            for (j = 1; j <= n; ++j)
            {
                if (board[i] == j)
                    printf("\tQ");
                else
                    printf("\t-");
            }
        }
        if(count==1)
        {
            exit(0);
        }
    }
    int main()
    {
        int n, i, j,c=0;
        void queen(int row, int n);
        printf(" - N Queens Problem Using Backtracking -");
        printf("\n\nEnter number of Queens:");
        fflush(stdin);
        scanf("%d", &n);
        counter(1,n,&c);
        printf("\nThe number of solutions possible: %d",c);
        queen(1, n);
        return 0;
    }
```