

NQUEENS

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

```
int n, count=0;
```

```
int isSafe(char board[n][n], int row, int col)
```

```
{
```

```
    for (int i = row - 1; i >= 0; i--)
```

```
    {
```

```
        if (board[i][col] == 'Q')
```

```
        {
```

```
            return 0;
```

```
        }
```

```
    }
```

```
    for (int i = row - 1, j = col - 1; i >= 0 && j >= 0; i--, j--)
```

```
    {
```

```
        if (board[i][j] == 'Q')
```

```
        {
```

```
            return 0;
```

```
        }
```

```
    }
```

```
    for (int i = row - 1, j = col + 1; i >= 0 && j < n; i--, j++)
```

```
    {
```

```
        if (board[i][j] == 'Q')
```

```
        {
```

```
            return 0;
```

```
        }
```

```
    }
```

```
    return 1;
```

```
}
```

```

void printBoard(char board[][n])
{
    printf("\nChess Board\n");

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("%c ", board[i][j]);
        }
        printf("\n");
    }
}

```

```

void nQueens(char board[n][n], int row)
{
    if (row == n)
    {
        printBoard(board);
        count++;
        return;
    }

    for (int j = 0; j < n; j++)
    {
        if (isSafe(board, row, j) == 1)
        {
            board[row][j] = 'Q';
            nQueens(board, row + 1);
            board[row][j] = 'X';
        }
    }
}

```

```

int main()

```

```

{
    printf("Enter the size of the board: ");
    scanf("%d", &n);

    char board[n][n];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            board[i][j] = 'X';
        }
    }
    nQueens(board, 0);
    printf("\nTotal Possible Solution: %d ",count);
}

```

OUTPUT:

```

Enter the size of the board: 4

Chess Board
X Q X X
X X X Q
Q X X X
X X Q X

Chess Board
X X Q X
Q X X X
X X X Q
X Q X X

Total Possible Solution: 2
Process returned 0 (0x0)   execution time : 1.259 s
Press any key to continue.

```