

EXP12

23UAI313

```
#include<iostream>
```

```
using namespace std;
```

```
#define N 4
```

```
void printSolution(int board[N][N]) {
```

```
    cout<<"Solution:\n";
```

```
    for (int i=0;i<N;i++)
```

```
    {
```

```
        for (int j=0;j<N;j++)
```

```
        {
```

```
            cout<<board[i][j]<<" ";
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    cout<<endl;
```

```
}
```

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

```
    for(int i=0;i<col;i++)
```

```
    {
```

```
        if(board[row][i])
```

```
            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<N && j>=0;i++,j--)
```

```
    {
```

```
        if(board[i][j])
```

```

        return false;
    }
    return true;
}

void solveNQueensUtil(int board[N][N],int col,int&solutionCount) {
    if(col>=N){
        printSolution(board);
        solutionCount++;
        return;
    }
    for (int i = 0; i < N; i++){
        if (isSafe(board, i, col)) {
            board[i][col] = 1;
            solveNQueensUtil(board,col+1,solutionCount);
            board[i][col] = 0;
        }
    }
}

void solveNQueens()
{
    int board[N][N]={0};
    int solutionCount=0;
    solveNQueensUtil(board,0,solutionCount);
    cout<<"Total solutions: "<<solutionCount<< "\n";
}

int main()
{
    solveNQueens();
}

```