**BackTrack, O(N!)**

**Tested on: uva 167**

int n;

bool board[10][10];

bool **isSafe**(int row, int col)

{

*//left*

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

**if**(board[row][i]) **return** **false**;

*//upper left*

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

**if**(board[i][j]) **return** **false**;

*//bottom left*

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

**if**(board[i][j]) **return** **false**;

**return** **true**;

}

void **solveNQ**(int col)

{

**if**(col>=n) **return**; //solution found, do additional calc

**for**(int i=0;i<n;i++)

**if**(**isSafe**(i, col))

{

board[i][col]=**true**;

**solveNQ**(col+1);

board[i][col]=**false**;

}

**return**;

}

**CLR**(board);

**solveNQ**(0);