|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77 | **import** **java.util.\***;  Implementation of N Queen Problem  **public** **class** **N\_Queens** {  **private** **static** **boolean** **isSafe**(**int** row,**int** col,**char**[][] board){  **for**(**int** j=**0**;j<board.length;j++){  **if**(board[row][j]=='Q') **return** **false**;  }  **for**(**int** i=**0**;i<board.length;i++){  **if**(board[i][col]=='Q') **return** **false**;  }  **for**(**int** i=row,j=col;i>=**0** && j>=**0**;i--,j--){  **if**(board[i][j]=='Q') **return** **false**;  }  **for**(**int** i=row,j=col;i>=**0** && j<board.length;i--,j++){  **if**(board[i][j]=='Q') **return** **false**;  }  **for**(**int** i=row,j=col;i<board.length && j>=**0**;i++,j--){  **if**(board[i][j]=='Q') **return** **false**;  }  **for**(**int** i=row,j=col;i<board.length && j<board.length;i++,j++){  **if**(board[i][j]=='Q') **return** **false**;  }  **return** **true**;  }  **private** **static** **void** **saveBoard**(**char**[][] board,List<List<String>> allBoards){  String str="";  List<String> newBoard = **new** ArrayList();  **for**(**int** i=**0**;i<board.length;i++){  String row="";  **for**(**int** j=**0**;j<board[**0**].length;j++){  **if**(board[i][j]=='Q'){  row+='Q';  }  **else**{  row+='\_';  }  }  newBoard.add(row);  }  allBoards.add(newBoard);  }  **private** **static** **void** **helper**(**char**[][] board,List<List<String>> allBoards,**int** col){  **if**(col==board.length){  saveBoard(board,allBoards);  **return**;  }  **for**(**int** row=**0**;row<board.length;row++){  **if**(isSafe(row,col,board)){  board[row][col]='Q';  helper(board,allBoards,col+**1**);  board[row][col]='\_';  }  }  }  **public** **static** List<List<String>> **solveNQueens**(**int** n) {  List<List<String>> allBoards=**new** ArrayList();  **char**[][] board =**new** **char**[n][n];  helper(board,allBoards,**0**);  **return** allBoards;  }  **public** **static** **void** **main**(String[] args) {  **int** n=**8**;  List<List<String>> ll=solveNQueens(n);  **int** i=**0**;  **for**(List<String> **res:**ll){  System.out.println("Sol."+(++i));  **for**(String **str:**res){  **for**(**char** **c:**str.toCharArray()){  System.out.print(c+" ");  }  System.out.println();  }  System.out.println('\n');  }  }  } |