```
#include <bits/stdc++.h>

using namespace std;

int board[11][11];

bool isPossible(int n,int row,int col){

// Same Column
  for(int i=row-1;i>=0;i--){
    if(board[i][col] == 1){
      return false;
    }
  }
//Upper Left Diagonal
  for(int i=row-1,j=col-1;i>=0 && j>=0 ; i--,j--){
    if(board[i][j] ==1){
      return false;
    }
  }

  // Upper Right Diagonal

  for(int i=row-1,j=col+1;i>=0 && j<n ; i--,j++){
    if(board[i][j] == 1){
      return false;
    }
  }

  return true;
}
void nQueenHelper(int n,int row){
  if(row==n){
    // We have reached some solution.
    // Print the board matrix
    // return

    for(int i=0;i<n;i++){
      for(int j=0;j<n;j++){
        cout << board[i][j] << " ";
      }
    }
    cout<<endl;
    return;

  }

  // Place at all possible positions and move to smaller problem
```

```
  for(int j=0;j<n;j++){

    if(isPossible(n,row,j)){
      board[row][j] = 1;
      nQueenHelper(n,row+1);
      board[row][j] = 0;
    }
  }
  return;

}
void placeNQueens(int n){

  memset(board,0,11*11*sizeof(int));

  nQueenHelper(n,0);

}
int main(){

  placeNQueens(4);
  return 0;
}
```