NIHARIKA AGRAWAL
R177219122
500075359
AIML BATCH 4

N-QUEEN PROBLEM

CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
void solveColumn(int);
bool isValidPlace(int , int);
void displayBoard();
void solveNqueen();

//Global variables
int queens;
int chessBoard[20][20];
bool hasSolution = false;

int main()
{
   //Enter value of N
   printf("Enter number of Queens \n");
   scanf("%d", &queens);
   solveNqueen();
   return 0;
}

void solveNqueen(){
   //starting from (0,0) of the board
   solveColumn(0);
```

NIHARIKA AGRAWAL
R177219122
500075359
AIML BATCH 4

```
   if(!hasSolution)
      printf("No Solution \n");
}

//Finding all solution by recusrive backtracking
void solveColumn(int col){

   //Reached beyond last column
   //Means solution (configuration) found
   if(col == queens){
      hasSolution = true;
      displayBoard();
      //intentionally returning to find more possible solution
      return;
   }

   for(int i=0; i<queens; i++){
      //checking if position is safe
      if(isValidPlace(i,col)){
         //setting current value to 1 means placing a queen
         chessBoard[i][col] = 1;
         //moving to next column's 1st row
         solveColumn(col+1);
         //backtrack - reset to 0 means removing queen
         chessBoard[i][col] = 0;
      }
   }
}
```

NIHARIKA AGRAWAL
R177219122
500075359
AIML BATCH 4

```
//To check whether the particular position is safe or not
bool isValidPlace(int row, int col){

   //Checking horizontally
   for(int i=col; i>=0; i--){
      if(chessBoard[row][i] == 1)
         return false;
   }

   //checking left diagonal
   for(int i=row, j=col; i>=0 && j>=0; i--,j--){
      if(chessBoard[i][j] == 1)
         return false;
   }

   //checking right diagonal
   for(int i=row, j=col; i<queens && j>=0; i++,j--){
      if(chessBoard[i][j] == 1)
         return false;
   }

   return true;
}

//Display chess board with queen configuration
void displayBoard(){
   for(int i=0; i<queens; i++){
```

NIHARIKA AGRAWAL
R177219122
500075359
AIML BATCH 4

```
      for(int j=0; j<queens; j++){
         if(chessBoard[i][j] == 1)
            printf(" * ");
         else
            printf(" - ");
      }
      printf("\n");
   }
    printf("\n\n");
}
```

OUTPUT