

# 数据结构课程设计（四）

## 项目文档

梁琛

1652770

同济大学软件学院

## 一、项目内容：

八皇后问题是一个古老而著名的问题，是回溯算法的经典问题。该问题是十九世纪著名的数学家高斯在 1850 年提出的：在  $8 \times 8$  的国际象棋棋盘上，安放 8 个皇后，要求没有一个皇后能够“吃掉”任何其它一个皇后，即任意两个皇后不能处于同一行，同一列或者同一条对角线上，求解有多少种摆法。

高斯认为有 76 种方案。1854 年在柏林的象棋杂志上不同的作者发表了 40 种不同的解，后来有人用图论的方法得到结论，有 92 中摆法。

本项目拓展了 N 皇后问题，即皇后个数由用户输入。

## 二、项目功能：

用户输入皇后个数 N。采用回溯法，首先将第一个皇后放置在第一行第一列，然后，依次在下一行上放置一个皇后，直到 N 个皇后全部放置安全。在放置每个皇后时，都依次对每一列进行检测，首先检测放在第一列是否与已放置的皇后冲突，如不冲突，则将皇后放置在该列，否则，选择改行的下一列进行检测。如整行的每一列都冲突，则回到上一行，重新选择位置。最后得到 N 皇后问题的所有解答。

## 三、用户手册：

用户根据系统提示输入 N 皇后问题中皇后的个数，如图 1：

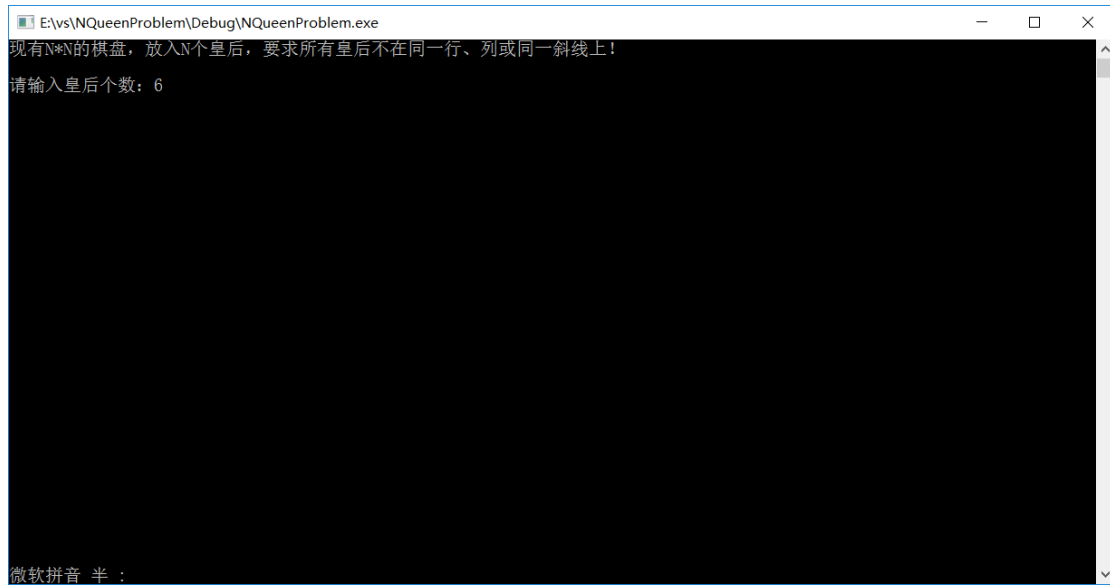


图 1

输入完毕后，系统会打印出所有可能的情况，并统计 N 皇后的个数，如图 2

( a ) 和图 2 ( b ):

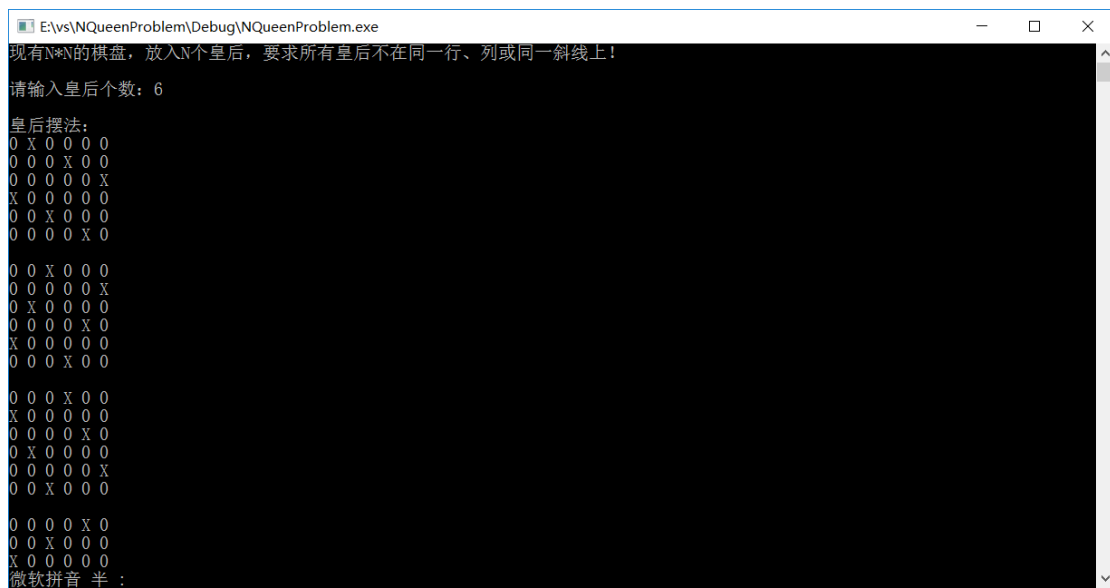


图 2 ( a )

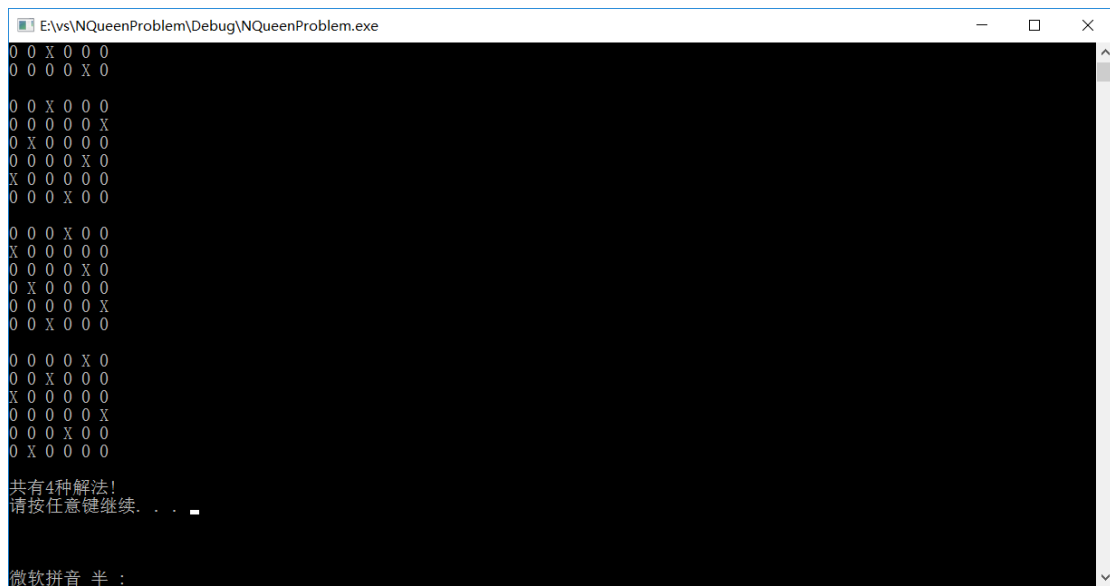


图 2 ( b )

#### 四、NQueenProblem 类（主系统类）接口：

函数名	参数	功能
NQueen(void)		递归方法求取皇后放置方案
getSum()		获得方案总数
Place(int r)	当前行的行号	检查参数所指示的这一行皇后放置方案是否满足要求
TraceBack(int r)	当前行的行号	用户递归求取皇后放置方案的递归方法
print()		打印当前成功的放置方案