**N皇后问题**

1. 项目简介：

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\*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。

本实验拓展了N皇后问题，即皇后个数由用户输入。

1. 项目功能要求:

八皇后在棋盘上分布的各种可能的格局数目非常大，约等于2的32次方种，但是，可以将一些明显不满足问题要求的格局排除掉。由于任意两个皇后不能同行，即每行只能放置一个皇后，因此将第i个皇后放在第i航上，这样在放置第i个皇后时，只要考虑它与前i-1个皇后处于不同列和不同对角线位置上即可。

解决这个问题采用回溯法，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，直到八个皇后全部放置安全。在放置每个皇后时，都依次兑每一列进行检测，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。如整行的八列都冲突，则回到上一行，重新选择位置，依次类推。

1. 具体实现：

1.数据结构：

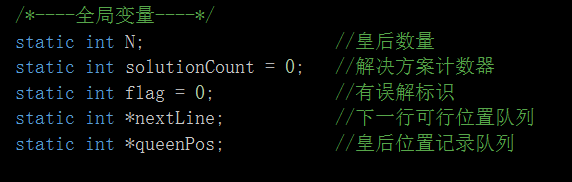
1.static int N //皇后数量

2.static int solutionCount; //解决方案计数器

3.static int flag //有误解标识

4.static int \*nextLine //下一行可行位置队列

5.staic int \*queenPos //皇后位置记录队列



2.函数

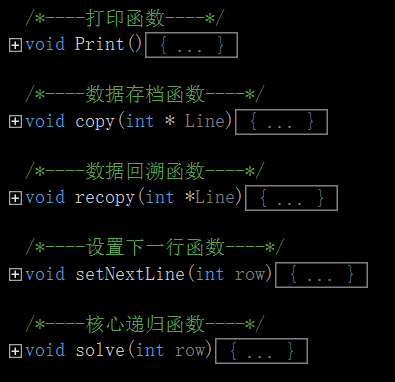
1.void Print() //打印可行序列

2.void vopy(int \* Line) //数据存档函数

3.void recopy(int \* Line) //回溯数据函数

4.void setNextLine(int row) //设置下一行不可行位置

5.void solve(int row) //核心递归函数

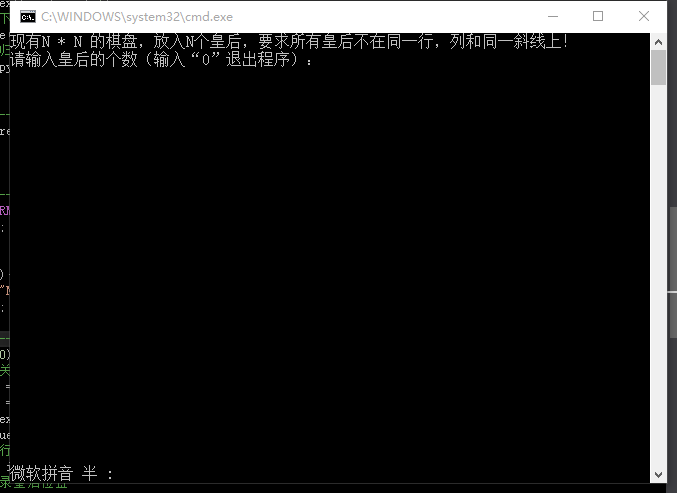


1. 算法分析：

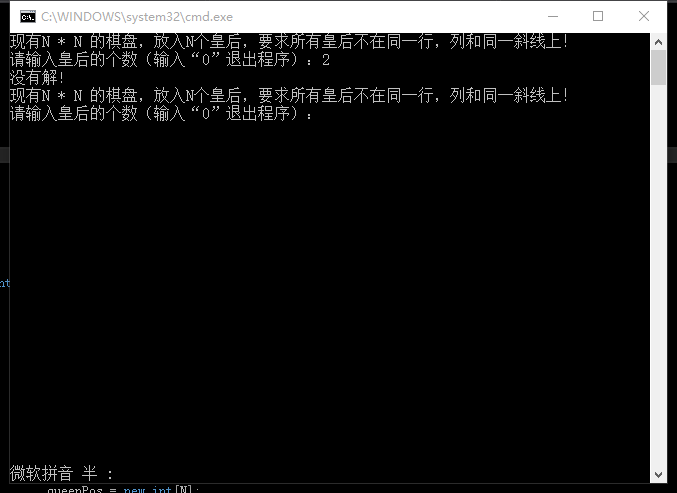
本题基本思路与第三题的迷宫基本相同，运用的是回溯思想。先将第一个皇后放入棋盘第一行，对第一行的每一个位置作一次循环即可得到所有结果。递归设置下一行可行位置，下一行判断可行位置，再对可行的位置作循环，每次进入递归都必须保存当前状态，一旦无法得到可行解，层层返回，然后回复当前状态，再看下一可行位置，直到第一行的皇后结束所有遍历。与勇闯迷宫的算法基本相似。

1. 程序截图：

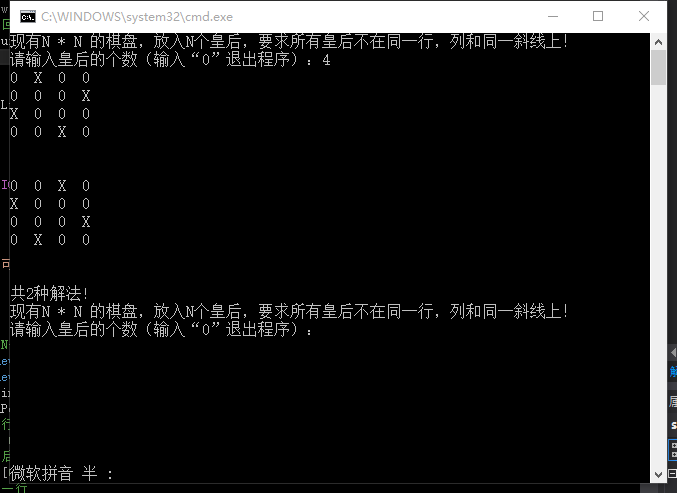
开始界面：



无解情况：



有解情况：





1. 体会：

这个项目考查的还是回溯法，并不是很难。这个项目让我深切体会到了运用计算机可以解决很多我们在数学上很难解决的问题，并对计算机有了更为浓厚的兴趣。