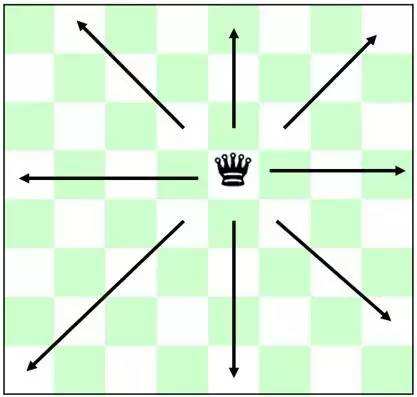
**八皇后问题**

一、背景

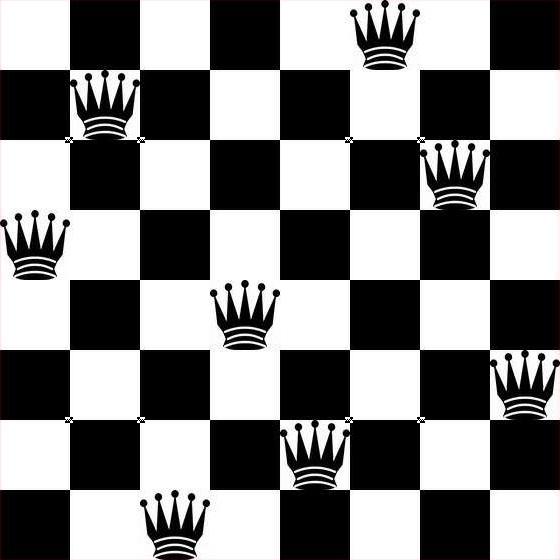
八皇后问题，是一个古老而著名的问题，是回溯算法的典型案例。

该问题是国际西洋棋棋手马克斯·贝瑟尔于1848年提出：在8×8格的国际象棋上摆放八个皇后，使其不能互相攻击，即任意两个皇后都不能处于同一行、同一列或同一斜线上，问有多少种摆法。 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解出92种结果**。**

(a)皇后的攻击范围：



(b)一个可行解：

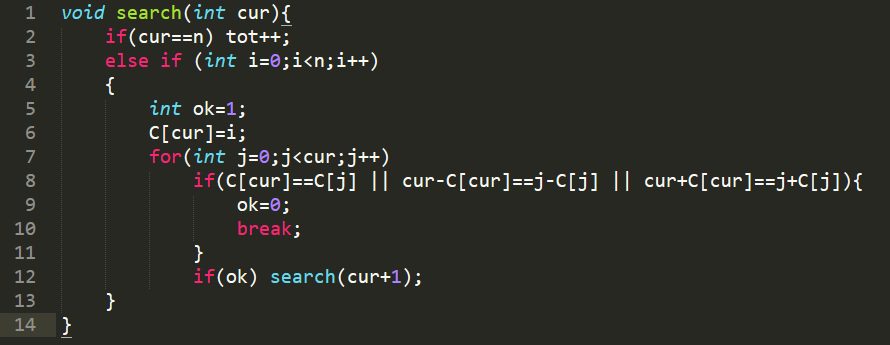


二、分析

思路1.最简单的思路是把问题转化为“从64个格子中选一个子集”，使得“子集中恰好有8个格子，且任意两个选出的格子都不在同一行、同一列或同一个对角线上”。这正是子集枚举问题。然而，64个格子的子集有2^64个，太大了。

思路2.把问题转化为“从64个格子里选出8个格子”，这是组合生成问题。根据高中组合数学知识，有=4.426\*10^9种方案，比第一种思路优秀，但仍然不够好。

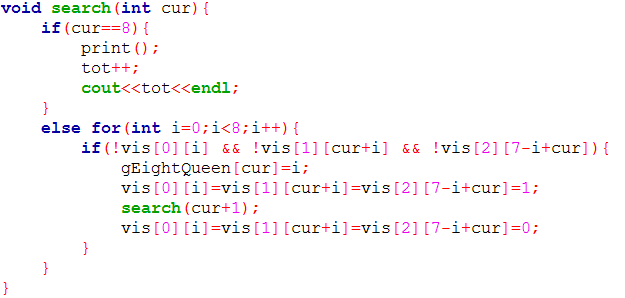
思路3.经过思考，发现：恰好每行每列各放置一个皇后。如果用C[x]表示第x行皇后的列编号，则问题变成了全排列生成问题。而0-7的排列一共有8！=40320个，枚举量不会超过它。



因为是逐行放置的，所以皇后不会横向攻击。只需要检查是否纵向和斜向攻击即可。

条件“cur-C[cur]==j-C[j]||cur+C[cur]==j+C[j]”用来判断皇后(cur,C[cur])和（j,C[j]）是否在同一对角线上。

思路4。程序效率可以继续提高：利用二维数组vis[3][15]直接判断当前尝试的皇后所在的列和两个对角线是否已有其他皇后。注意到主对角线标识y-x可能为负，存取时要加上7.



源程序：

