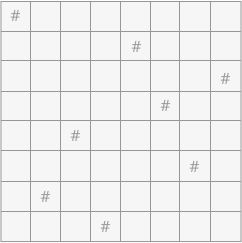
 八皇后问题是以国际象棋为背景的问题：有八个皇后（可以当成八个棋子），如何在 8\*8 的棋盘中放置八个皇后，使得任意两个皇后都不在同一条横线、纵线或者斜线上。



代码：

#include <stdio.h>

int Queenes[8]={0},Counts=0;

int Check(int line,int list){

//遍历该行之前的所有行

for (int index=0; index<line; index++) {

//挨个取出前面行中皇后所在位置的列坐标

int data=Queenes[index];

//如果在同一列，该位置不能放

if (list==data) {

return 0;

}

//如果当前位置的斜上方有皇后，在一条斜线上，也不行

if ((index+data)==(line+list)) {

return 0;

}

//如果当前位置的斜下方有皇后，在一条斜线上，也不行

if ((index-data)==(line-list)) {

return 0;

}

}

//如果以上情况都不是，当前位置就可以放皇后

return 1;

}//输出语句

void print()

{ for (int line = 0; line < 8; line++)

{

int list;

for (list = 0; list < Queenes[line]; list++)

printf("0");

printf("#");

for (list = Queenes[line] + 1; list < 8; list++){

printf("0");

}

printf("\n");

}

printf("================\n");

}

void eight\_queen(int line){

//在数组中为0-7列

for (int list=0; list<8; list++) {

//对于固定的行列，检查是否和之前的皇后位置冲突

if (Check(line, list)) {

//不冲突，以行为下标的数组位置记录列数

Queenes[line]=list;

//如果最后一样也不冲突，证明为一个正确的摆法

if (line==7) {

//统计摆法的Counts加1

Counts++;

//输出这个摆法

print();

//每次成功，都要将数组重归为0

Queenes[line]=0;

return;

}

//继续判断下一样皇后的摆法，递归

eight\_queen(line+1);

//不管成功失败，该位置都要重新归0，以便重复使用。

Queenes[line]=0;

}

}

}

int main() {

//调用回溯函数，参数0表示从棋盘的第一行开始判断

eight\_queen(0);

printf("摆放的方式有%d种",Counts);

return 0;

}

基本思路是：  
1.第一行先占一个皇后  
2.第二行再占一个且不能与第一个皇后攻击  
3.第三行再占一个  
。。。。。  
n.第n行占一个，当第n行站不下的时候，取消n-1行的皇后，在第n-1皇后的下一个位置重新占一个皇后位置，知道占到最n-1行的最后一个位置，当还不行的时候，就取消第n-2行，当n-2行的皇后在n-2行的最后一个位置的时候，就取消n-3，n-2在最后一个位置，那么n-3行的一定不再最后一个位置。  
再重新寻找n-2行的皇后的位置。

。。。。  
。。。。  
直到找到最后一个皇后。