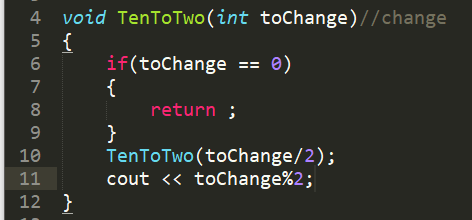
**第三章 栈和队列**

1. 3.12非负十进制整数转二进制

对于非负整数n，如果n等于0，输出0，结束；如果n大于0，n%2得到的结果x(x==1,0)是n转化成二进制的最后一位数字，对于rem=n/2（相当于左移一位），重复如上过程，可得到之前的二进制位。由于转换时先得到最后的二进制位，所以先调用递归函数，然后输出该位二进制数。

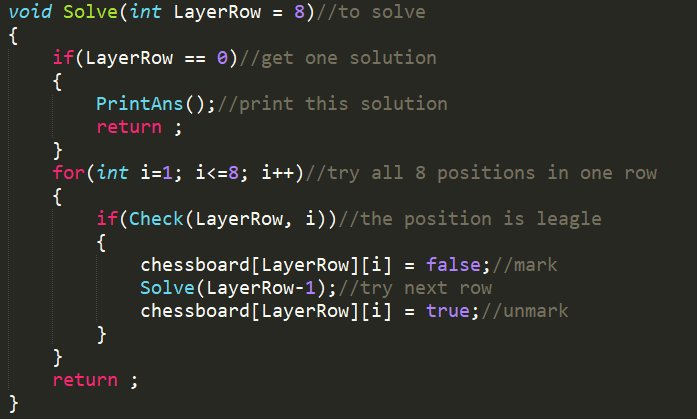
主要代码如下：



1. 3.17八皇后问题

八皇后问题可分为八层，每一行为一层。通过调用递归函数，层次依次增加，对于每一层，穷举该行每一列的八个点，然后检查是否与前面的皇后冲突，如果不冲突，递归下一层，直到达到第九层，递归到达出口，输出该组答案。

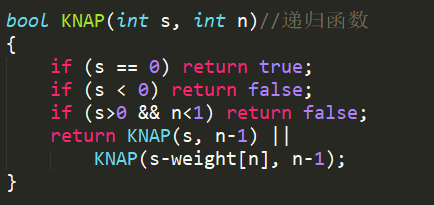
主要代码如下：



1. 3.18背包问题

根据书上给出的定义，直接写出递归函数。当背包容量S等于0时，正好填满背包，返回True。如果背包小于0，无法填满，返回FALSE。如果背包还有空间，但是已经取完所有物品，返回FALSE。否则返回是否去该物品的问题。

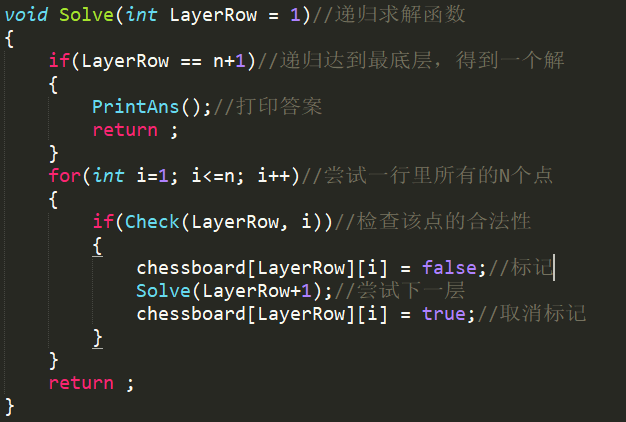
主要代码如下：



1. 3.19 n个数字全排列

相比于八皇后问题，去掉八皇后中斜对角线上会攻击的条件，每行每列的元素唯一，以每一行为一层，每一列表示所取的数字，就可以得到8个数字的全排列，将8改成n，得到n个数的全排列。

主要代码如下：



1. 3.20 n个数字取r个组合

同样类比于八皇后，把棋盘改成n行r列的棋盘，可以实现n个数字取出r个的组合。为了保证组合不重复，每一次递归所取的数字必须大于上一层取的数字。

主要代码如下：

