[Catalan数](http://blog.csdn.net/acdreamers/article/details/7628667)

**Catalan数的定义：**

设表示用下面的方法把凸多边形区域分成三角形区域的方法数：在有n+1条边的凸多边形区域内通过插入在其中不相交的对角线而把它分成三角形区域。定义。则满足递推关系

这个递推关系的解是：，这里的叫做**Catalan数**。

那么上面的递推式的正确性我们可以简单描述一下即可：

**证明：**这里因为表示按照上述规则划分的三角形区域个数，那么我们随便选一条多边形的一条边作为基边，那么

     再在剩余的n-1个点中选一个点，我们把所选的一条边的两点分别与所选的那一点连接起来，那么多边形被划

     分成3部分，一部分有k+1条边，一部分有3条边，另一部分有n-k+1条边，那么这样就划分成了子问题了，所

     以按照这个思路可以证明递推式成立。

那么根据递推式是如何推出Catalan数的通项公式呢？

这里用到了生成函数：我们很容易写出的生成函数

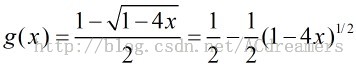
我们进一步计算



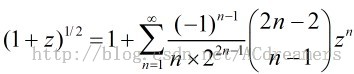
因为有：，所以进一步得到：

，由于

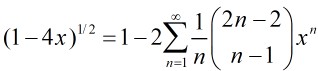
所以有：，解之得到：

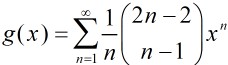
，另一个解不符合，舍去。

那么根据牛顿二项式有：



那么带入化简得到：



那么我们最终得到：

所以：，这就是**Catalan的推导过程**。

卡特兰数的应用

**1、括号化问题**

矩阵连乘：，依据乘法结合律，不改变其顺序，只用括号表示成对的乘积，问有几种括号化的方案？

**2、出栈次序问题**

一个栈(无穷大)的进栈序列为1,2,3,..n,有多少个不同的出栈序列?

**类似问题**

a、有2n个人排成一行进入剧场，入场费5元。其中只有n个人有一张5元钞票，另外n人只有10元钞票，剧院无其它钞票，问有多少中方法使得只要有10元的人买票，售票处就有5元的钞票找零？(将持5元者到达视作将5元入栈，持10元者到达视作使栈中某5元出栈)

b、n个1和n个0组成一个2n位的二进制数，要求从左到右扫描，0的累计数不小于1的累计数，求满足条件的的数。

c、12个人排成两排，每排必须是从矮到高排列，而且第二排比对应的第一排的人高，问排列方式有多少种？

我们先把这12个人从低到高排列,然后,选择6个人排在第一排,那么剩下的6个肯定是在第二排.用0表示对应的人在第一排,用1表示对应的人在第二排,那么含有6个0,6个1的序列,就对应一种方案.

比如000000111111就对应着

第一排：0 1 2 3 4 5 第二排：6 7 8 9 10 11 010101010101就对应着 第一排：0 2 4 6 8 10 第二排：1 3 5 7 9 11问题转换为，这样的满足条件的01序列有多少个。与情况b一样。

**3、给定节点组成二叉树的问题**

给定N个节点，能构成多少种形状不同的二叉树？

先取一个点作为顶点,然后左边依次可以取0至N-1个相对应的,右边是N-1到0个,两两配对相乘,就是h(0)\*h(n-1) + h(2)\*h(n-2) +  + h(n-1)h(0)=h(n))能构成h（N）个

**4.n\*n棋盘从左下角走到右上角而不穿过主对角线的走法**

**5.n个+1和n个-1构成的2n项序列，其部分和总满足：的序列的个数。**

Catalan数的高精度处理：利用递归式： h(n)=((4\*n-2)/(n+1))\*h(n-1)

1. **int** a[105][105];    //大数卡特兰数
2. **int** b[105];         //卡特兰数的长度
4. **void** catalan()  //求卡特兰数
5. {
6. **int** i,j,len,carry,temp;
7. a[1][0]=b[1]=1;
8. len=1;
9. **for**(i=2;i<=100;i++)
10. {
11. **for**(j=0;j<len;j++)    //乘法
12. a[i][j]=a[i-1][j]\*(4\*(i-1)+2);
13. carry=0;
14. **for**(j=0;j<len;j++)    //处理相乘结果
15. {
16. temp=a[i][j]+carry;
17. a[i][j]=temp%10;
18. carry=temp/10;
19. }
20. **while**(carry)    //进位处理
21. {
22. a[i][len++]=carry%10;
23. carry/=10;
24. }
25. carry=0;
26. **for**(j=len-1;j>=0;j--) //除法
27. {
28. temp=carry\*10+a[i][j];
29. a[i][j]=temp/(i+1);
30. carry=temp%(i+1);
31. }
32. **while**(!a[i][len-1])     //高位零处理
33. len--;
34. b[i]=len;
35. }
36. }
38. **int** main()
39. {
40. **int** i,n;
41. catalan();
42. **while**(~scanf("%d",&n),n)
43. {
44. **for**(i=b[n]-1;i>=0;i--)
45. printf("%d",a[n][i]);
46. printf("\n");
47. }
48. **return** 0;

}