如果要把药吃完，那么这n片要就需要去n次半片，n次整片。令取整片的次数标记为1，取半片的个数为0，从左往右扫描1的个数必须大于等于0的个数。这样就转化为了一个卡特兰数问题

仔细想想可以把问题转换一下，最后肯定2n长度的序列，肯定是n个完整药片和n个半个药片。

并且当前半个药片的数量不超过 前面完整药片的数量！

所以：

可以转换成卡特兰数列！

状态转移：

dp[i][j] = dp[i-1][j] + dp[i][j-1];

排列数量中来源，i-1完整，j个不完整，和  i个完整，j-1个不完整！

最后dp[n][n]就是答案！

#include<cstdio>

#include<cstring>

#include<algorithm>

using namespace std;

const int maxn = 100 + 10;

typedef long long ll;

ll dp[maxn][maxn];

void init(){

for (int i = 0; i <= 30; ++i)dp[i][0] = 1;

for (int i = 1; i <= 30; ++i)

for (int j = 1; j <= i; ++j)

dp[i][j] = dp[i-1][j] + dp[i][j-1];

}

int main(){

int n;

init();

while(scanf("%d",&n) == 1 && n){

printf("%I64d\n",dp[n][n]);

}

return 0;

}

或卡特兰数

#include<stdio.h>

//打表卡特兰数

//第 n个 卡特兰数存在a[n]中，a[n][0]表示长度；

//注意数是倒着存的，个位是 a[n][1] 输出时注意倒过来。

int a[105][100];

void ktl()

{

int i,j,yu,len;

a[2][0]=1;

a[2][1]=2;

a[1][0]=1;

a[1][1]=1;

len=1;

for(i=3;i<101;i++)

{

yu=0;

for(j=1;j<=len;j++)

{

int t=(a[i-1][j])\*(4\*i-2)+yu;

yu=t/10;

a[i][j]=t%10;

}

while(yu)

{

a[i][++len]=yu%10;

yu/=10;

}

for(j=len;j>=1;j--)

{

int t=a[i][j]+yu\*10;

a[i][j]=t/(i+1);

yu = t%(i+1);

}

while(!a[i][len])

{

len--;

}

a[i][0]=len;

}

}

int main()

{

ktl();

int n;

while(~scanf("%d",&n) && n)

{

for(int i=a[n][0];i>0;i--)

{

printf("%d",a[n][i]);

}

puts("");

}

return 0;

}