## 2032：杨辉三角

### 一、题目

#### 问题描述

还记得中学时候学过的杨辉三角吗？具体的定义这里不再描述，你可以参考以下的图形： 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1

#### 输入数据

输入数据包含多个测试实例，每个测试实例的输入只包含一个正整数n（1<=n<=30），表示将要输出的杨辉三角的层数。

#### 输出数据

对应于每一个输入，请输出相应层数的杨辉三角，每一层的整数之间用一个空格隔开，每一个杨辉三角后面加一个空行。

#### 输入样例

2 3

#### 输出样例

1

1 1

1

1 1

1 2 1

#### 题目来源

HDU 2032 http://acm.hdu.edu.cn/showproblem.php?pid=2032

### 二、题解

#### 解题思路

**杨辉三角**大家应该都不陌生。我们知道从第二行起，每行首尾两项均为1，且从第三行起，除去首尾两项的每一项（i , j）（r为行号，c为列号)，均满足（i，j）=（i-1，j）+（i-1，j-1），由此我们可以构建一个由杨辉三角组成的二维数组，再根据给定的输入，输出对应的前n行即可。

注意：输出每次行完成后,都要进行换行，到一下个三角形之前也要换行，记得每个数字最后没有空格。

#### 参考程序

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int c[30][30];  
 for(int i=0; i<30; i++) //先打印杨辉三角  
 {  
 c[i][0]=1;  
 c[i][i]=1;  
 }  
 for(int i=2; i<30; i++)  
 for(int j=1; j<i; j++)  
 c[i][j]=c[i-1][j-1]+c[i-1][j];  
 int n;  
 while(scanf("%d",&n)!=EOF)  
 {  
 for(int i=0; i<n; i++) //按照要求打印杨辉三角  
 {  
 for(int j=0; j<=i; j++)  
 {  
 printf("%d",c[i][j]);  
 if(j==i)  
 printf("\n");  
 else  
 printf(" ");  
 }  
 if(i==n-1)  
 printf("\n");  
 }  
 }  
 return 0;  
}

#### 复杂度分析

由源代码可以看出该程序主要分为两部分：第一部分为614行，预处理了杨辉三角各行的值，运算了O（N2）次；第二部分为1830行，输出符合题设的杨辉三角，共有2层for循环，每层约做O (N2) 次.所以总的时间复杂度为O（N2+ N2）= O（N2）。

#### 编程技巧

（1） 使用二维数组储存杨辉三角，避免重复计算。