**杨辉三角形**

问题描述

杨辉三角形又称Pascal三角形，它的第i+1行是(a+b)i的展开式的系数。

它的一个重要性质是：三角形中的每个数字等于它两肩上的数字相加。

下面给出了杨辉三角形的前4行：

   1

  1 1

 1 2 1

1 3 3 1

给出n，输出它的前n行。

输入格式

输入包含一个数n。

输出格式

输出杨辉三角形的前n行。每一行从这一行的第一个数开始依次输出，中间使用一个空格分隔。请不要在前面输出多余的空格。

样例输入

4

样例输出

1  
1 1  
1 2 1  
1 3 3 1

数据规模与约定

1 <= n <= 34。

锦囊1

使用二维数组。

锦囊2

使用一个二维数组来保存杨辉三角形，先将三角形计算出来，然后再输出。

C++代码

#include <iostream>

using namespace std;

const int MAXN = 40;

int n;

int a[MAXN][MAXN];

int main()

{

cin >> n;

a[0][0] = 1;

for (int i = 0; i < n; ++i)

{

a[i][0] = a[i][i] = 1;

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

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

}

for (int i = 0; i < n; ++i)

{

for (int j = 0; j <= i; ++j)

cout << a[i][j] << " ";

cout << endl;

}

return 0;

}

C代码

#include <stdio.h>

#define MAXN 40

int n;

int a[MAXN][MAXN];

int main()

{

int i, j;

scanf("%d", &n);

a[0][0] = 1;

for (i = 0; i < n; ++i)

{

a[i][0] = a[i][i] = 1;

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

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

}

for (i = 0; i < n; ++i)

{

for (j = 0; j <= i; ++j)

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

printf("\n");

}

return 0;

}

Java代码

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

public class Main{

public static void main(String args[]) throws IOException{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int n=Integer.parseInt(br.readLine());

int[][] a=new int[34][34];

int i=0,j=1;

while(i<34){

a[i][0]=1;

i++;

}

while(j<34){

a[j][j]=1;

j++;

}

for(i=2;i<34;i++)

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

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

for(i=0;i<n;i++)

{for(j=0;j<=i;j++)

System.out.print(a[i][j]+" ");

System.out.println(" ");}

}

}