



## 打印数字直角三角形

### 【问题描述】

输入一个正整数 $n$ ，其中 $n \leq 9$ 。

输出一个 $n$ 行的数字直角三角形。

第一行是 1，第二行是 12，第三行是 123……

### 【输入样例】

6

### 【输出样例】

```
1
12
123
1234
12345
123456
```

### 【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << j;
        }
        cout << endl;
    }
    return 0;
}
```



## 打印循环数字直角三角形

### 【问题描述】

输入一个正整数 $n$ 。

输出一个 $n$ 行的数字直角三角形。

其中第一行为数字 1，第二行为数字 23，第三行为数字 456，第四行为数字 7890，第 5 行为 12345，……

### 【输入样例】

6

### 【输出样例】

```
1
23
456
7890
12345
678901
```

### 【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int t = 0;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            t = (t + 1) % 10;
            cout << t;
        }
        cout << endl;
    }
    return 0;
}
```



## 打印杨辉直角三角形

### 【问题描述】

输入一个正整数 $n$ 。

输出一个 $n$ 行的杨辉直角三角形。

每个数字的输出宽度为 6。

### 【输入样例】

6

### 【输出样例】

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

### 【算法分析】

每行的第一个数是 1。

迭代公式：

$$p = p \times \frac{i-j}{j}$$

$i$  是行号,  $j$  是列号。

### 【参考程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
```



```
{  
    int p = 1;  
    cout << p;  
    for(int j = 1; j < i; j++)  
    {  
        p = p * (i - j) / j;  
        cout << setw(6) << p;  
    }  
    cout << endl;  
}  
return 0;  
}
```



## 打印杨辉三角形

### 【问题描述】

输入一个正整数 $n$ 。

输出一个 $n$ 行的杨辉三角形。

每个数字的输出宽度为 6。

### 【输入样例】

6

### 【输出样例】

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

### 【算法分析】

每行的第一个数是 1。

迭代公式：

$$p = p \times \frac{i-j}{j}$$

$i$  是行号,  $j$  是列号。

### 【参考程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
```



```
{
    int p = 1;
    cout << setw(3 * (n - i) + 1) << p;
    for (int j = 1; j < i; j++)
    {
        p = p * (i - j) / j;
        cout << setw(6) << p;
    }
    cout << endl;
}
return 0;
}
```