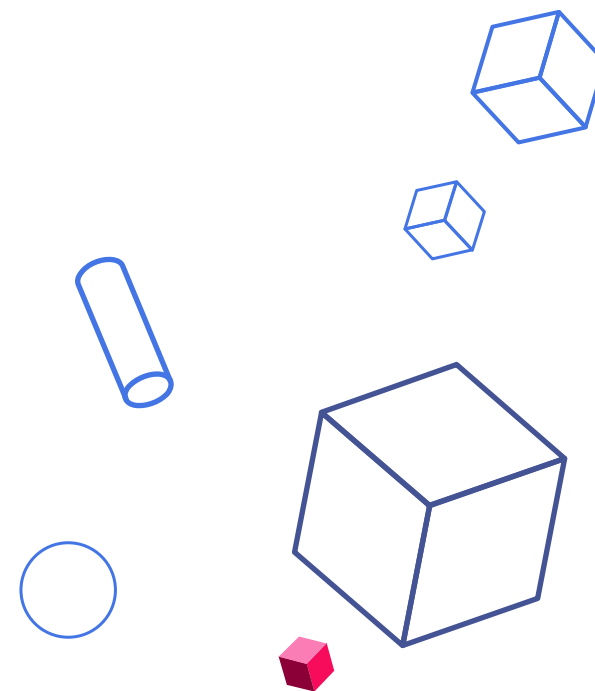


00 习题讲解



例1：角谷猜想

编写一程序，验证角谷猜想。

所谓的角谷猜想是："对于任意大于1的自然数 n ，若 n 为奇数，则将 n 变为 $3*n+1$ ，否则将 n 变为 n 的一半。经过若干次这样的变换，一定会使 n 变为1。

输入
大于1的自然数 n
输出
 n 值改变过程

样例输入
7
样例输出
22 11 34 17 52 26
13 40 20 10 5 16
8 4 2 1

分析

样例输入

7

样例输出

22 11 34 17 52 26

13 40 20 10 5 16

8 4 2 1

模拟过程:

第一次: $n=7$ 奇数 $n=3*n+1$; $n=22$

第二次: $n=22$ 偶数 $n=n/2$; $n=11$

第三次: $n=11$ 奇数 $n=3*n+1$; $n=34$

...

第...次: $n=1$ 结束

Q:用for循环还是while循环?

次数不定, 用while合适

每次循环:

- 先判断 n 的奇偶性
- 根据奇偶性改变 n 的值, 输出 n 的值
- 继续下一次循环

所需变量: n

循环条件: $n!=1$

循环体: $\text{if}(n\%2==1)\{$

$n=3*n+1;$
输出 n

$\}$

$\text{else}\{$

$n=n/2;$
输出 n

$\}$

语句块

分析

核心代码：

```
int n;  
cin>>n;  
while( n!=1 ){  
    if(n%2==1){  
        n=3*n+1;  
        cout<<n<<" ";  
    }  
    else{  
        n=n/2;  
        cout<<n<<" ";  
    }  
}
```

所需变量： n

循环条件： $n \neq 1$

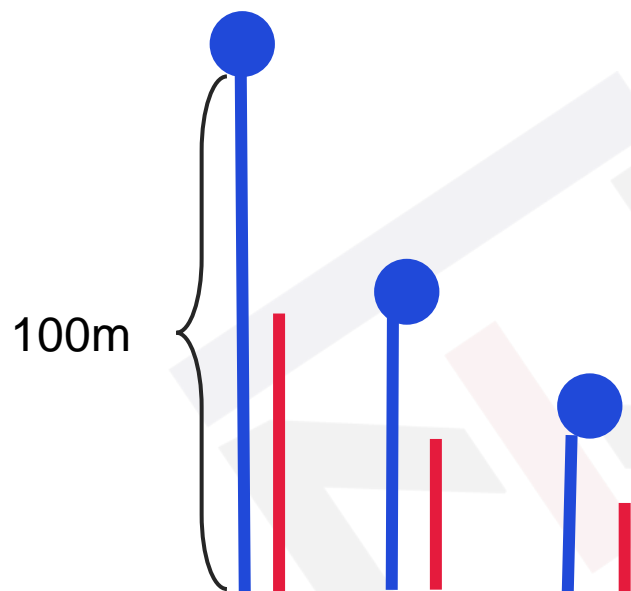
循环体： $\text{if}(n \% 2 == 1) \{$
 $n = 3 * n + 1;$
 输出n
 $\}$
 else {
 $n = n / 2;$
 输出n
 }

循环次数由是否达到条件决定，while循环也被称为**条件循环**

例2：小球下落

一球从100米高度自由下落，每次落地后反跳会原来高度的一半，再落下，求他在第10次落地时，共经过多少米？第10次反弹多高？

分析



设总路程为s 高度为h

分别的初始值: $s=0, h=100$

第1次下落、反弹: 100m 50m $s=s+100+50$ $h=50m$

第2次下落、反弹: 50m 25m $s=s+50+25$ $h=25m$

第3次下落、反弹: 25m 12.5m $s=s+25+12.5$ $h=12.5m$

...

直到第n次下落、反弹

每次下落h、反弹h/2 $s=s+h+h/2;$

下一次的高度减半 $h=h/2;$

循环体

核心代码

```
int i;
double s,h;
s=0;
h=100;
for(i=1;i<=10;i=i+1){
    s=s+h+h/2;
    h=h/2;
}
printf("%.6lf %.6lf",s,h);
```

299.707

循环体部分:

每次下落h、反弹h/2 $s=s+h+h/2;$
下一次的高度减半 $h=h/2;$

总路程s和高度h的初始值:

$s=0;$ $h=100;$

循环变量初始值及条件:

$i=1;$ $i\leq 10$

核心代码

```
int i;
double s,h;
s=0;
h=100;
for(i=1;i<=10;i=i+1){
    s=s+h+h/2;
    h=h/2;
}
printf("%.6lf %.6lf",s,h);
```

299.707

循环体里出现了两次除法

计算机二进制除法转换为十进制时会有精度丢失的问题

Q: 如果减少除法的次数, 得到精度更高的答案?

2019年3月7日 BigInteger bigSub = big1.subtract(big2); //multiply实现乘法运算 BigInteger bigMul = big1.multiply(big2); //divide实现除法运算 BigInteger big...

博客园 百度快照

[float除法精度损失-CSDN论坛](#)

10条回复 - 发帖时间: 2010年7月31日

2010年7月31日 float及double类型减法运算时精度丢失问题 当float和double类型在进行减法运算时,会出现精度丢失问题,这种问题主要是由于计算机中普遍使用2进制所造...

CSDN技术社区 百度快照

[请教:为了运算速度快用了整数除法但精度丢失怎么办_\(amob...](#)

5条回复 - 发帖时间: 2016年4月8日

2016年4月8日 请教:为了运算速度快用了整数除法但精度丢失怎么办,有什么简练和算法吗? 同步轮周长"传动比/步进每转脉冲数"1000000 同步轮周长=齿数"齿距(5.08mm) 10...

www.amobbs.com/thread-5647706-... 百度快照

[关于运算时的精度丢失问题-CSDN论坛](#)

3条回复 - 发帖时间: 2013年9月29日

大家都知道float运算时,可能会存在精度丢失的问题,例如 1/3 * 3 不等于1,但是我有一个问题希望大家帮助解答,如下是我的测试...

CSDN技术社区 百度快照

[计算机在作浮点数运算时,在哪些情况下可能丢失数据精度 - ...](#)

1个回答 - 回复时间: 2016年4月19日

最佳答案: 十进制数的二进制表示形式可能不精确,使用的数字之间类型不匹配(例如,混合使用浮点型和双精度型),为解决此行为,大多数程序员或是确保值比...

更多关于计算机除法精度丢失的问题>>

百度知道 百度快照

[Java - 如何避免在除法和转换为int期间丢失精度? - IT屋...](#)

2018年8月6日 您可以使用它"精确"乘以5和1.2得到6。 本文地址:IT屋 » Java - 如何避免在除法和转换为int期间丢失精度?java casting int precision ...

www.it1352.com/9070...html 百度快照

分析

设总路程为 s 高度为 h

假定小球已经下落了100m, 起点从底部开始
分别的初始值: $s=100, h=50$

第1次反弹=第2次下落

第2次反弹=第3次下落

第3次反弹=第4次下落

第 n 次反弹=第 $n+1$ 次下落

每次下落 h 、反弹 h

$$s=s+2*h$$

下一次的高度减半

$$h=h/2;$$

循环体

核心代码

```
int i;  
double s,h;  
s=100;  
h=50;  
for(i=2;i<=10;i=i+1){  
    s=s+2*h;  
    h=h/2;  
}  
printf("%.6lf %.6lf",s,h);
```

299.609

循环体部分:

每次下落h、反弹h $s=s+2h$
下一次的高度减半 $h=h/2$;

总路程s和高度h的初始值:

$s=100$; $h=50$;

循环变量初始值及条件:

$i=2$; $i\leq 10$

小结

```
int i;  
double s,h;  
s=0;  
h=100;  
for(i=1;i<=10;i=i+1){  
    s=s+h+h/2;  
    h=h/2;  
}  
printf("%.6lf %.6lf",s,h);
```

299.707

```
int i;  
double s,h;  
s=100;  
h=50;  
for(i=2;i<=10;i=i+1){  
    s=s+2*h;  
    h=h/2;  
}  
printf("%.6lf %.6lf",s,h);
```

299.609

虽然以上两种方法在数学上无任何差距
但是由于第二种方法更符合计算机的计算方式，能得到精度更高的结果。

例3：斐波那契数列

斐波那契数列，又称黄金分割数列或兔子数列，指的是这样一个数列：1、1、2、3、5、8、13、21、.....输出前n项。

样例

输入

5

输出

1 1 2 3 5

分析

1、1、2、3、5、8、13、21

模拟一下每一项如何计算：

$$1+1=2$$

$$1+2=3$$

$$2+3=5$$

...

得到规律：

第n项=第n-2项+第n-1项

$$c=a+b$$

c

a

b

分析

第1次: 1、1、2、3、5、8、13、21
a b c

第2次: 1、1、2、3、5、8、13、21
a b c

第3次: 1、1、2、3、5、8、13、21
a b c

每次计算完c以后, 更新a, b的值

分析

循环体部分：

- 算出新的c
- 输出本次计算的c
- 本次循环的b等于下一次循环的a
- 本次循环的c等于下一次循环的b

循环变量初始值及循环条件：

$i=3;$ $i \leq n$

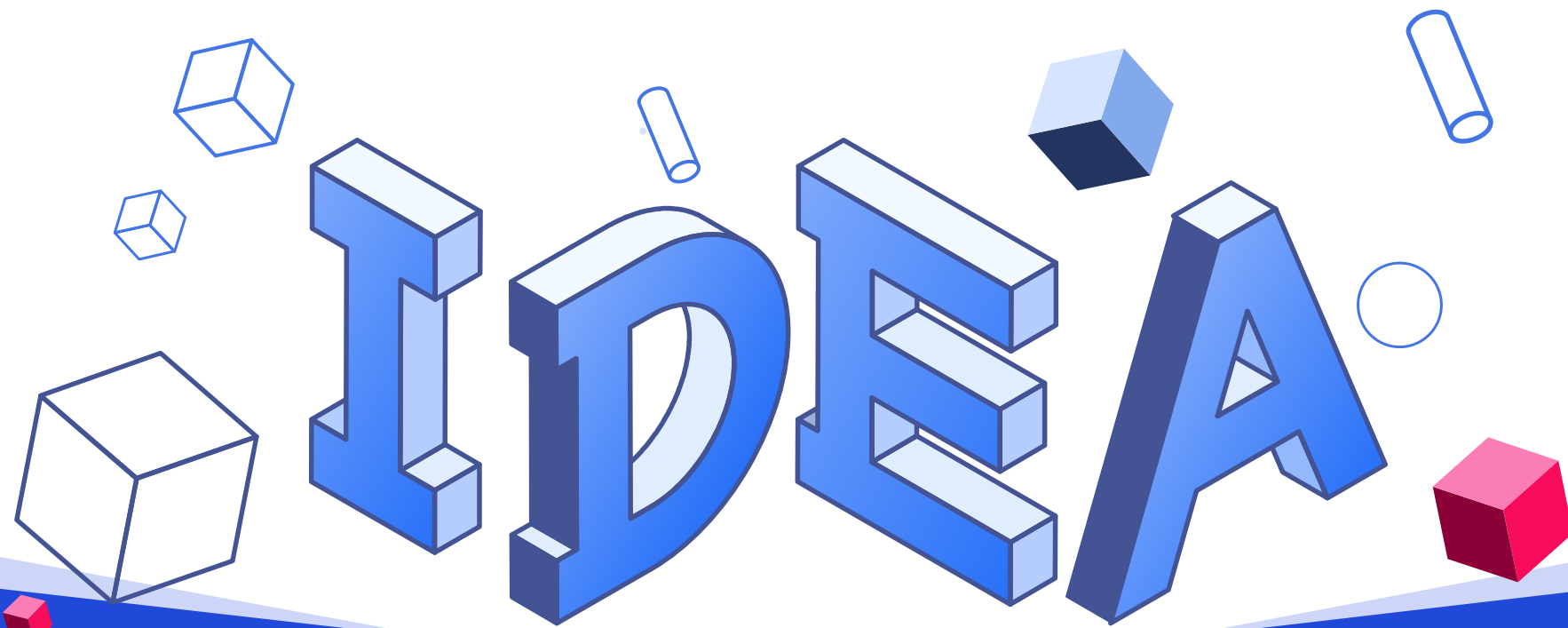
哪些项数是无法计算的？

第一项和第二项 a,b初始值为1

Q：如果 $n=1$ 或 2 ，循环是否能计算？

核心代码

```
int a,b,c,n,i;
cin>>n;
a=1;
b=1;
if(n==1)    //需要特别考虑n==1或2
    cout<<1;
else {
    cout<<1<<" "<<1<<" ";
    for( i=3; i<=n; i=i+1 ) {
        c=a+b;
        cout<<c<<" ";
        a=b;
        b=c;
    }
}
```

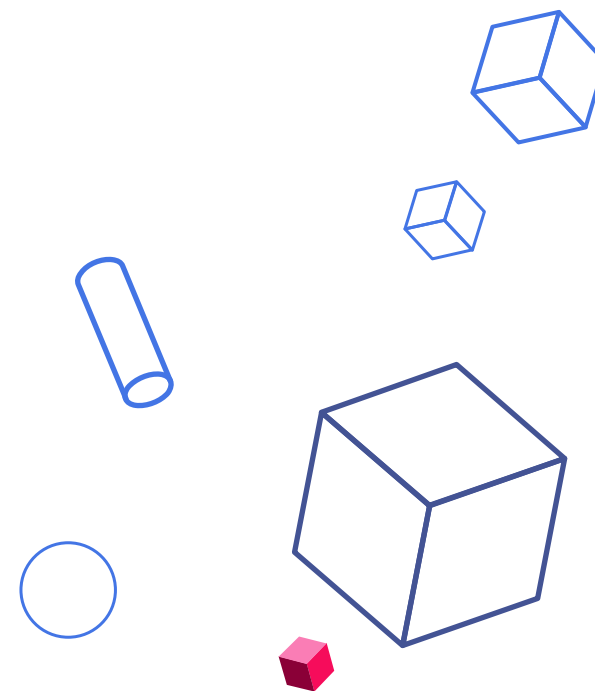



信息思维课程

循环嵌套

2021

01 格式输出



例1：星号三角形

输入一个n，输出一个n行的星号三角形

观察下图的输出特点

- 1.输入不同的行数，查看程序的输出
- 2.观察行数与星号格式的关系，思考如何实现

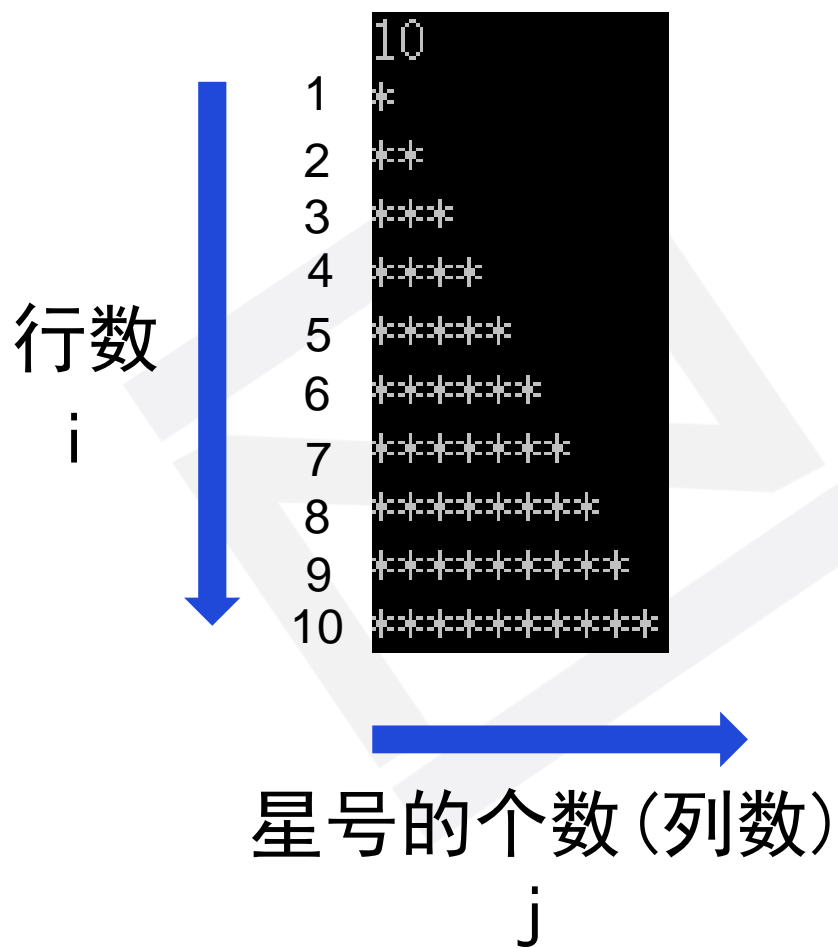
请输入星号三角形的行数：

4

*

**

分析

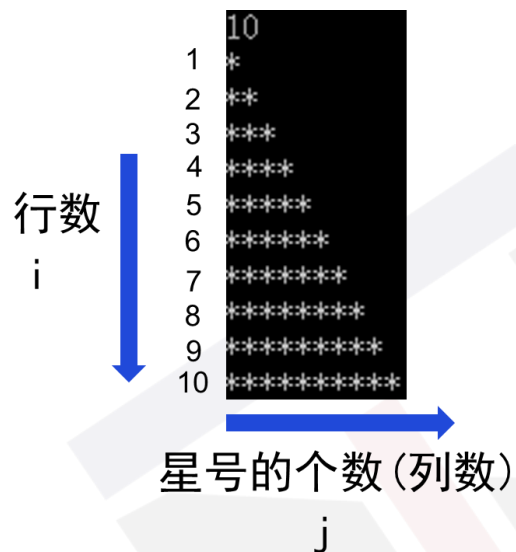


首先观察三角形的组成

1. 一共要输出n行 **第一层循环**
2. 第i行有i个星号* **第二层循环**

Q：一层循环能否进行输出？

分析



1. 定义变量

n, i, j

2. 循环变量初始化及终止条件

$i=1;$ $i \leq n$ 1. 一共要输出 n 行
 $j=1;$ $j \leq i$ 2. 第 i 行有 i 个星号*

3. 根据分析写出循环嵌套

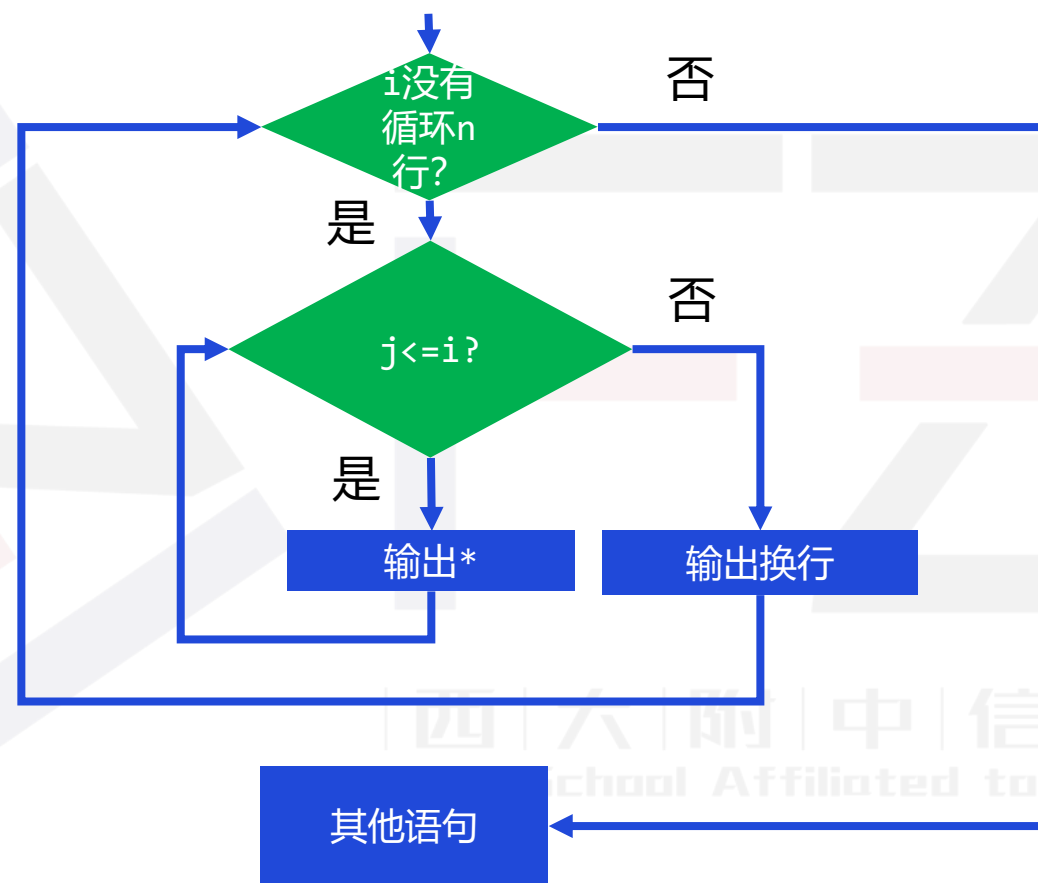
一共是两层循环
外层循环 i : 行数
内层循环 j : 列数

```
for( 行数  $i:1 \sim n$  ){  
    for( 列数  $j:1 \sim i$  ){  
        输出*;  
    }  
    输出换行  
}
```

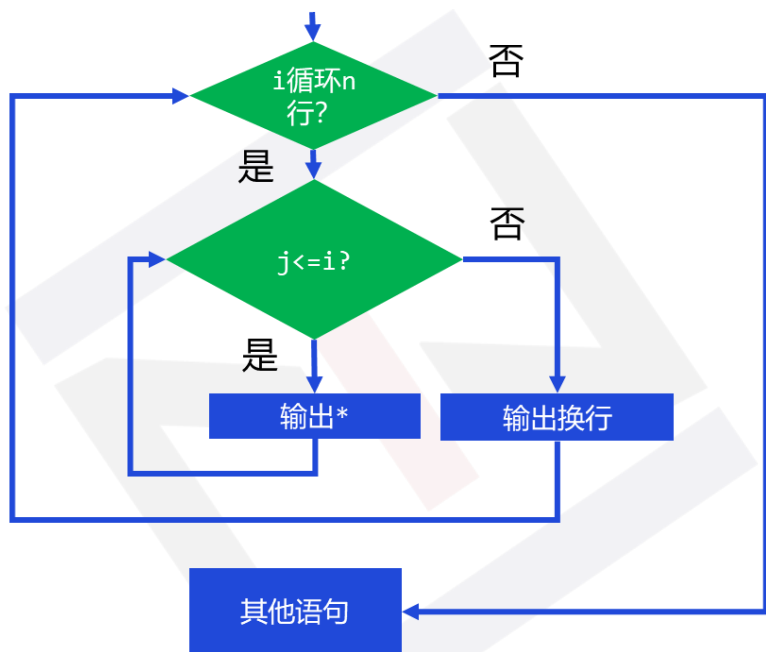
4. 分别完善对应的循环体

外层循环 { 输出 i 个* —— 内层循环 (循环 i 次) 输出*
输出换行

流程图



代码



```
#include <bits/stdc++.h>
using namespace std;
int main()
{
```

```
    int n,i,j;
    cin>>n;
    for(i=1;i<=n;i++){ //行数
        for(j=1;j<=i;j++){ //每一行星号个数
            cout<<"*";
        }
        cout<<endl;
    }
    return 0;
}
```

循环里还有循环称为循环嵌套

分析

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n,i,j;
    cin>>n;
    for(i=1;i<=n;i++){ //行数
        for(j=1;j<=i;j++){ //每一行星号个数
            cout<<"*";
        } //内层循环体
        cout<<endl;
    } //外层循环体
    return 0;
}
```

类似于甲、乙、丙、丁，多层循环嵌套时，循环变量命名约定俗成使用i,j,k,l,m,n

举例

```
for(i=1;i<=n;i=i+1){  
    for(j=1;j<=n;j=j+1){  
        for(k=1;k<=n;k=k+1){  
            .....  
        }  
    }  
}
```

例2：数字三角形

输入一个正整数 n ，输出 n 行数字三角形。其中，第1行为数字1，第2行为数字23，第3行为数字456，第4行为数字7890，第5行为数字12345

样例输入

5

样例输出

1

23

456

7890

12345

分析

1
23
456
7890
12345

与符号三角形类似

观察规律

1. 一共有 n 行，第 i 行有 i 个数 **循环嵌套**
2. 数字从1开始输出，到10则输出0，然后重新从1计数 **计数器+分支语句判断**

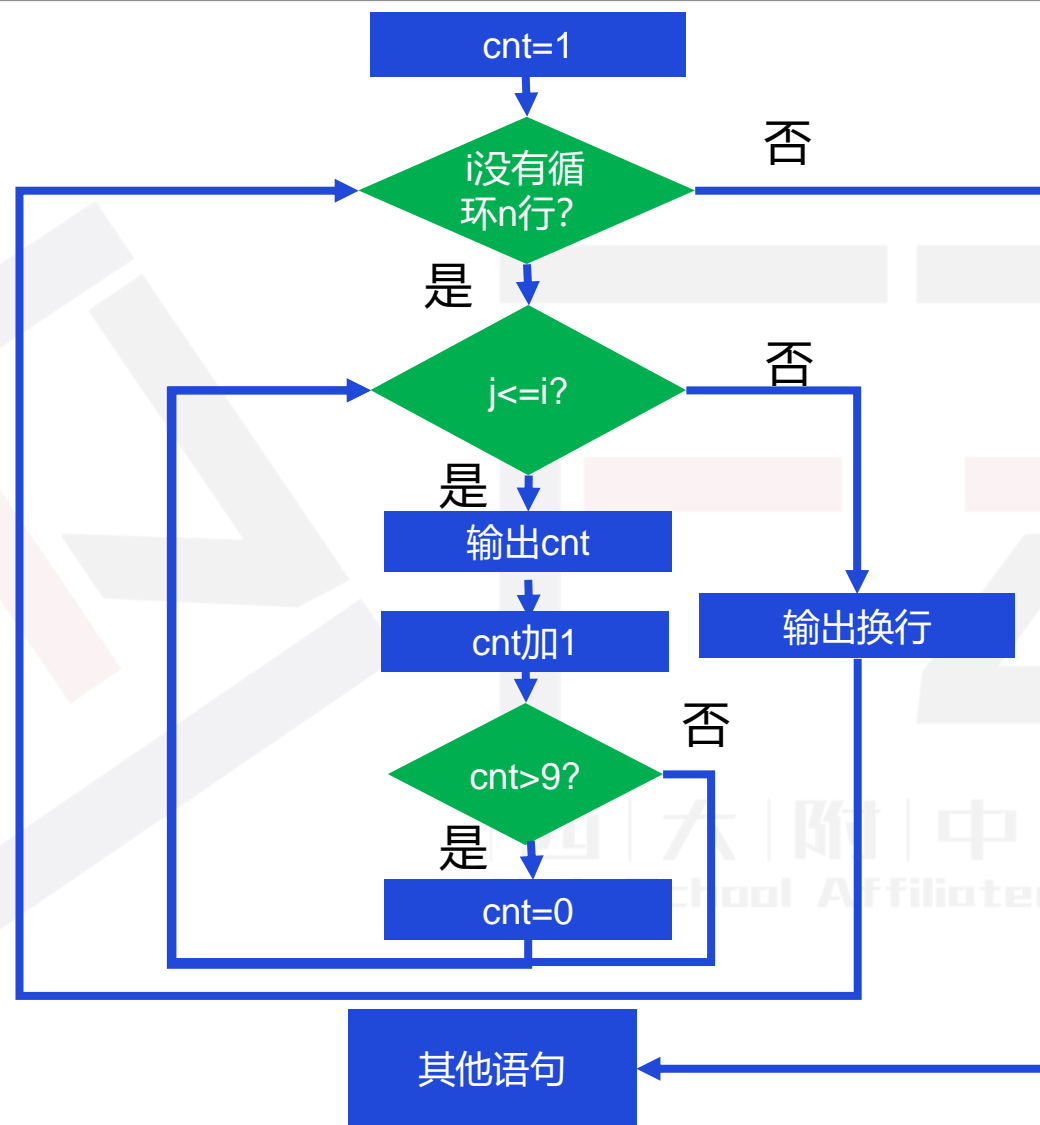
分析

1
23
456
7890
12345

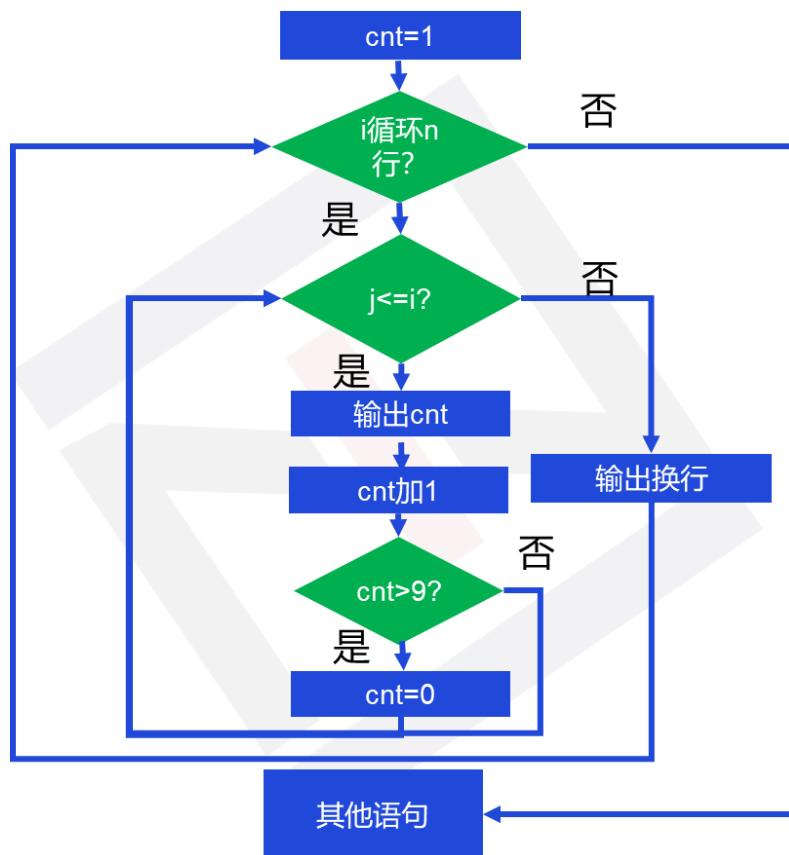
```
设置计数器cnt =1
for(i:1~n){ //控制行数
    for(j:1~i){ //每一行数字的个数
        //内层循环体
        输出计数器cnt的值
        计数器+1
        如果cnt>9
            计数器cnt=0;
    }
    cout<<endl; //每一行数字输出完之后，换行
}
```

西华大学附属中学
High School Affiliated to Southwest University

流程图



代码



```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,cnt,i,j; //cnt用于计数
```

```
    cin>>n;
```

```
    cnt=1;
```

```
    for(i=1;i<=n;i++){ //控制行数
```

```
        for(j=1;j<=i;j++){ //每一行数字的个数
```

```
            cout<<cnt;
```

```
            cnt=cnt+1;
```

```
            if(cnt>9)
```

```
                cnt=0;
```

```
        }
```

```
        cout<<endl; //每一行数字输出完之后，换行
```

```
    }
```

```
    return 0;
```

```
}
```

分析

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n,cnt,i,j; //cnt用于计数
    cin>>n;
    cnt=1;
    for(i=1;i<=n;i++){ //控制行数
        for(j=1;j<=i;j++){ //每一行数字的个数
            cout<<cnt;
            cnt=cnt+1;
            if(cnt>9)
                cnt=0;
        }
        cout<<endl; //每一行数字输出完之后，换行
    }
    return 0;
}
```

Q: 能否优化为一行代码?

提示: 使用%

取余运算

\rightarrow **$cnt=(cnt+1)\%10$**

取余运算

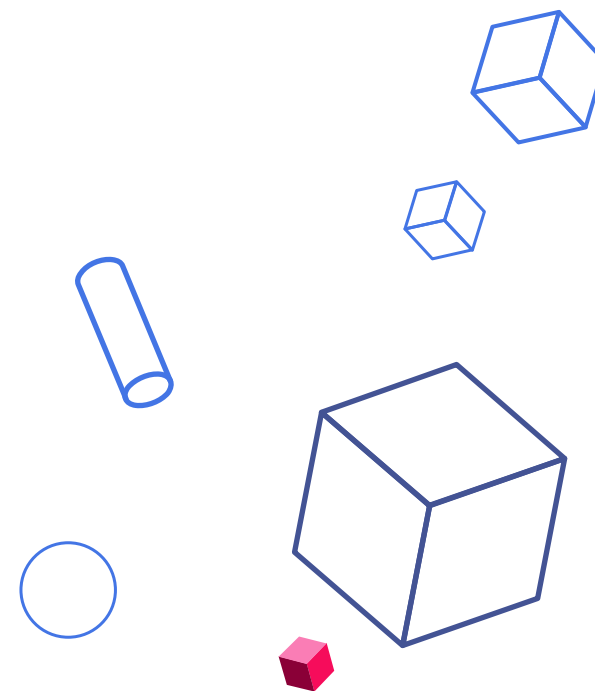
$8\%10=8$

$9\%10=9$

$10\%10=0$

启示: 常常可以通过一些数学运算或方法简化代码, 提升程序运行效率

02 枚举计算



例1：鸡兔同笼

鸡兔同笼问题：已知鸡和兔的总量为 n ，总腿数为 m ，输入 n 和 m ，依次输出鸡和兔的数目，如果无解，则输出 “No answer”。

输入
鸡和兔的总量 n ，总腿数 m 。

输出
依次输出鸡和兔的数目，如果无解，则输出 “No answer”。

样例
输入
10 20
输出
10 0

报告老师，这道题做过了



分析

在这之前，我们利用“全鸡全兔法”这样的数学方法解决了这道题目

设鸡的只数为a，兔为b

“全兔法”

$$a = (4 * n - m) / 2;$$
$$b = n - a;$$

“全鸡法”

$$b = (m - 2 * n) / 2;$$
$$a = n - b;$$

分析

但是，在刚开始学习的时候，我相信大家都想到过这样一种方法：



鸡

0只

1只

2只



兔

0只 、 1只 、 2只 ...

0只 、 1只 、 2只 ...

0只 、 1只 、 2只 ...

...

穷举

Q：是什么让你放弃了这种做法？

因为这样“拼凑”答案，对于人来说计算量大，无法高效的求解问题的答案
但计算机的运算速度是很快的，这样的计算量对于它来说“小菜一碟”

分析



鸡

列举鸡的只数

1只

2只

3只

行



兔

1只 、 2只 、 3只 ...

1只 、 2只 、 3只 ...

1只 、 2只 、 3只 ...

...

列

列举兔的只数

分析

算法

输入总的只数 n , 总的腿数 m

```
for(列举鸡的只数i: 0~n ){  
    for(列举兔的只数j: 0~n ){  
        if(满足条件){  
            输出当前的鸡和兔子的只数  
        }  
    }  
}
```



```
cin>>n>>m;  
for(i=0;i<=n;i=i+1){  
    for(j=0;j<=n;j=j+1){  
        if(i+j==n && 2*i+4*j==m){  
            cout<<i<<" "<<j<<endl;  
        }  
    }  
}
```

满足的条件: 总只数为 n $i+j==n$
 总腿数为 m $2*i+4*j==m$

分析

```
flag=0;
cin>>n>>m;
for(i=0;i<=n;i=i+1){
    for(j=0;j<=n;j=j+1){
        if(i+j==n&&2*i+4*j==m){
            cout<<i<<" "<<j<<endl;
            flag=1;
        }
    }
}
if(flag==0)
    cout<<"No answer"<<endl;
```

Q: 什么时候输出无解的情况?

循环没有穷举到答案

Q: 如何表示没有穷举到答案?

标记, flag

flag=0表示无解

flag=1表示有解

例2：百钱买百鸡

公鸡5元1只，母鸡3元1只，小鸡3只1元。现用100元买了100只鸡，编程求公鸡、母鸡、小鸡各买了多少只？（要求至少1只。）

输入
无

输出
一行一个解，每行3个数，分别表示公鸡个数，母鸡个数，
小鸡个数，中间以空格隔开。

样例
输入
无
输出
4 18 78
8 11 81
12 4 84

先思考一下如何穷举

分析



列举公鸡的数目

1只

2只

...



列举母鸡的数目

1只

2只

...

1只

2只

...



3、6、9...只

3、6、9...只

3、6、9...只

3、6、9...只

列举小鸡的数目

分析

算法

```
for(列举公鸡的只数i: 1~20 ){  
    for(列举母鸡的只数j: 1~33 ){  
        for(列举小鸡的只数k: 3~100 ){  
            if(满足条件){  
                输出当前的公鸡、母鸡、小鸡的只数  
            }  
        }  
    }  
}
```

满足的条件: 总只数为100 $i+j+k==100$
 总钱数为100 $5*i+3*j+k/3==100$

```
for(i=1;i<=20;i=i+1){  
    for(j=1;j<=33;j=j+1){  
        for(k=3;k<=100;k=k+3){  
            if( i+j+k==100 && 5*i+3*j+k/3==100){  
                cout<<i<<" "<<j<<" "<<k<<endl;  
            }  
        }  
    }  
}
```

分析

```
for(i=1;i<=20;i=i+1){  
    for(j=1;j<=33;j=j+1){  
        for(k=3;k<=100;k=k+3){  
            if(i+j+k==100&&5*i+3*j+k/3==100){  
                cout<<i<<" "<<j<<" "<<k<<endl;  
            }  
        }  
    }  
}
```

总的循环次数=i的次数*j的次数*k的循环次数

Q: 是否能够减少一层循环, 提高程序效率?

可以减少穷举小鸡的循环, 通过数学计算得到

$k=100-i-j$;

分析

优化后

```
for(i=1;i<=20;i=i+1){  
    for(j=1;j<=33;j=j+1){  
        k=100-i-j;  
        if( 5*i+3*j+k/3==100 &&k%3==0 ){  
            cout<<i<<" "<<j<<" "<<k<<endl;  
        }  
    }  
}
```

小结

对于人来说，穷举一个问题的答案，是很困难且浪费时间的

而通过借助计算机强大的运算速度以及不会出错的特性，可以通过多层循环穷尽问题的各种情况，在可接受的时间内找到问题最终的答案。

在程序设计中，这样的方法叫 **“穷举法”** (也叫枚举法)

But

但...

由于每道题目有

时间限制

1 秒

太过于暴力的穷举，往往会

测评结果

运行超时

欧，我的天！
怎么会这样



西南大学附属中学 | 信息学竞赛 |
High School Affiliated to Southwest University

Thanks

For Your Watching

