

第5章 循环控制

——嵌套循环

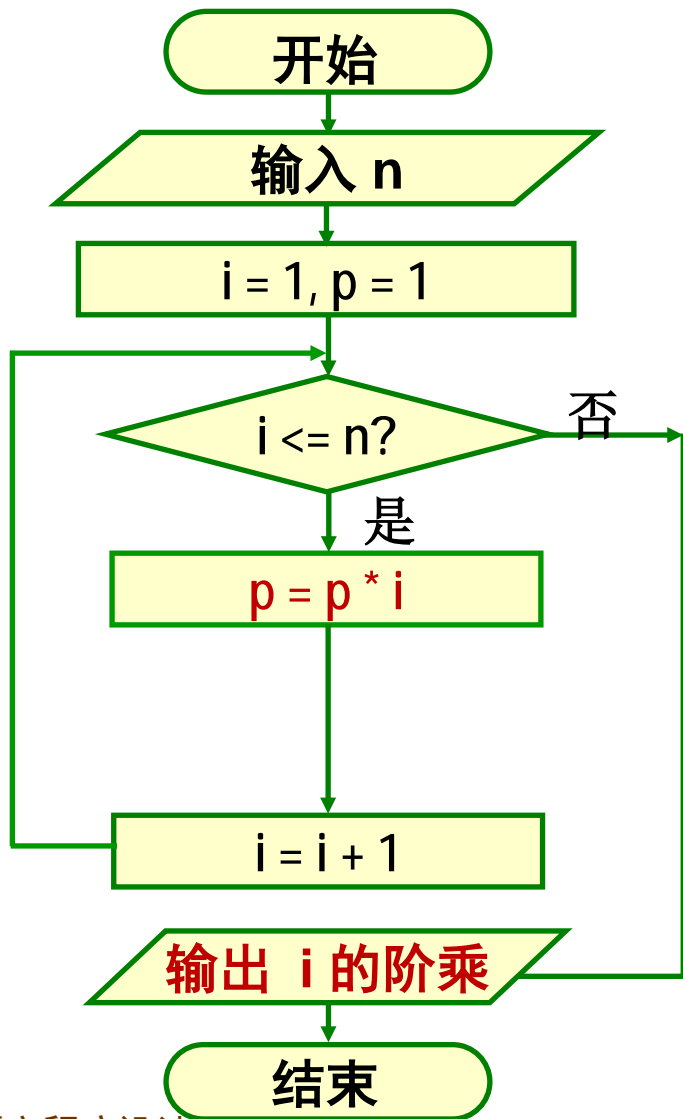
本节要讨论的主要问题

- 嵌套循环是如何执行的？
- 如何设计嵌套循环的程序？



循环实现累加累乘

计算并输出 $n!$

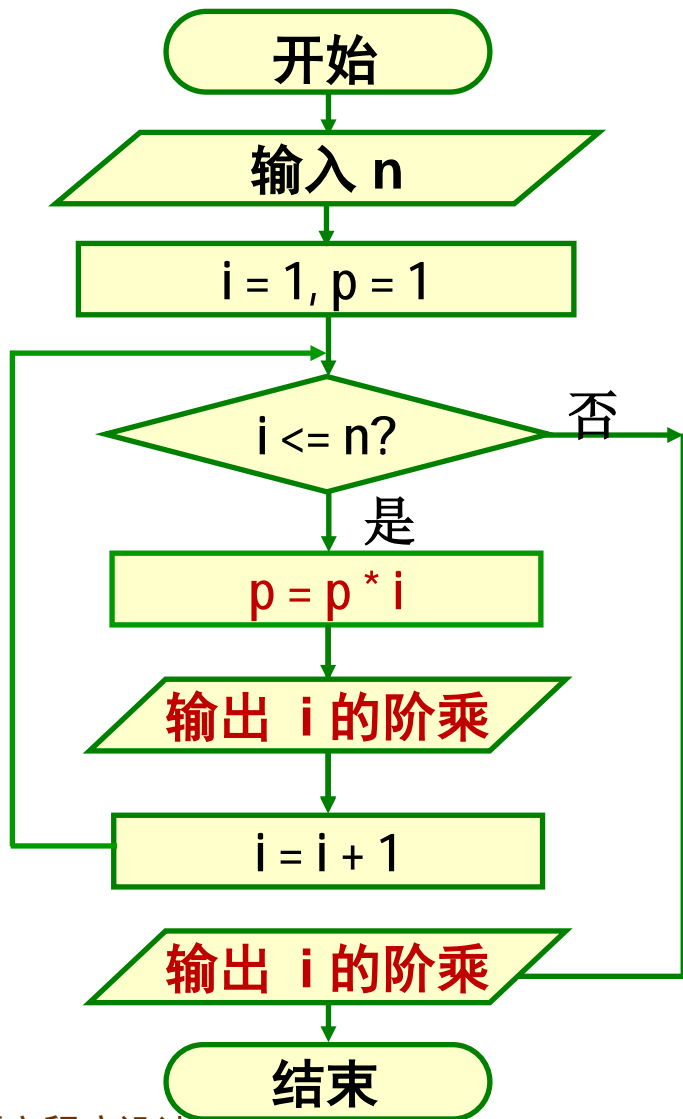


```
#include <stdio.h>
int main()
{
    int    i, n;
    long   p = 1;
    printf("Please enter n:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        p = p * i;
    }
    printf("%ld\n", p);
    return 0;
}
```

```
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9
```

循环实现累加累乘

计算并输出 $1!$, $2!$, $3!$, ..., $n!$

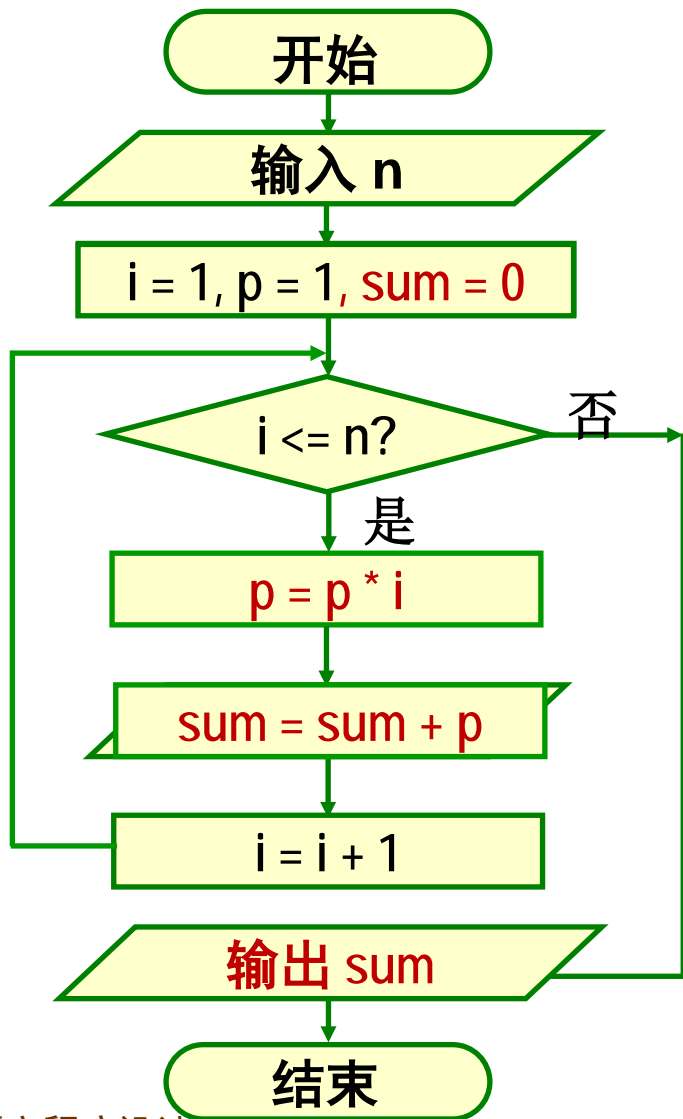


```
#include <stdio.h>
int main()
{
    int    i, n;
    long   p = 1;
    printf("Please enter n:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        p = p * i;
    }
    printf("%ld\n", p);
    return 0;
}
```

```
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9
```

循环实现累加累乘

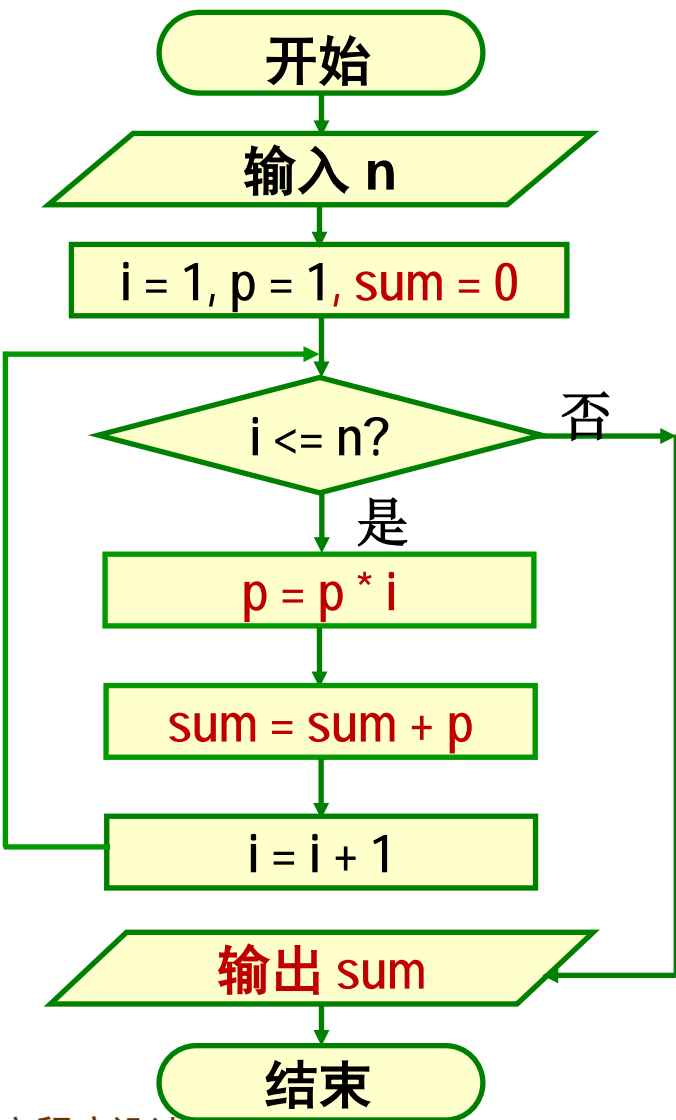
计算并输出 $1!, + 2!, + 3!, \dots, + n!$



```
#include <stdio.h>
int main()
{
    int    i, n;
    long   p = 1; sum = 0;
    printf("Please enter n:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        p = p * i;
        sum = sum + p;
    }
    printf("sum = %ld\n", sum);
    return 0;
}
```

```
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9
```

循环实现累加累乘



寻找累加项的构成规律

前后项
有关

- 利用前项计算后项
- $\text{sum} = \text{sum} + p$

前后项
无关

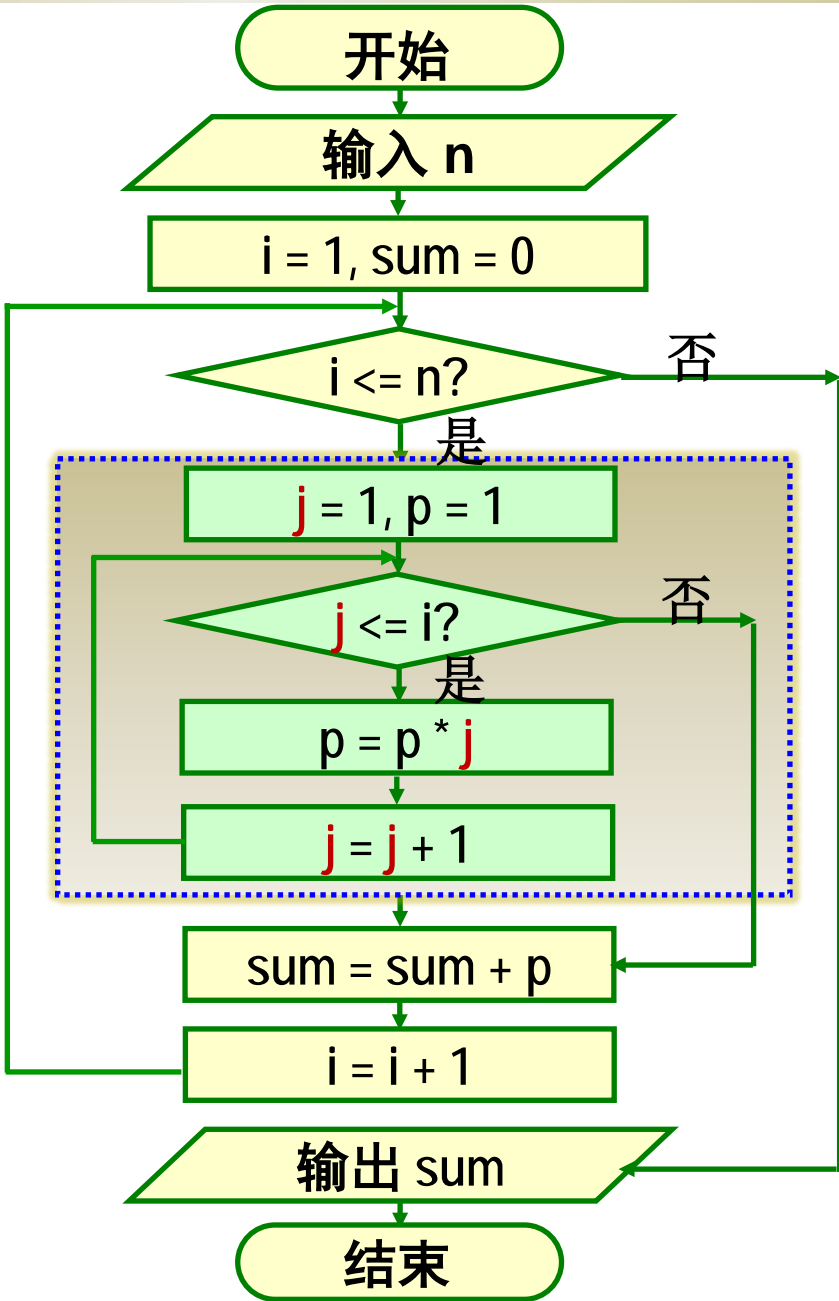
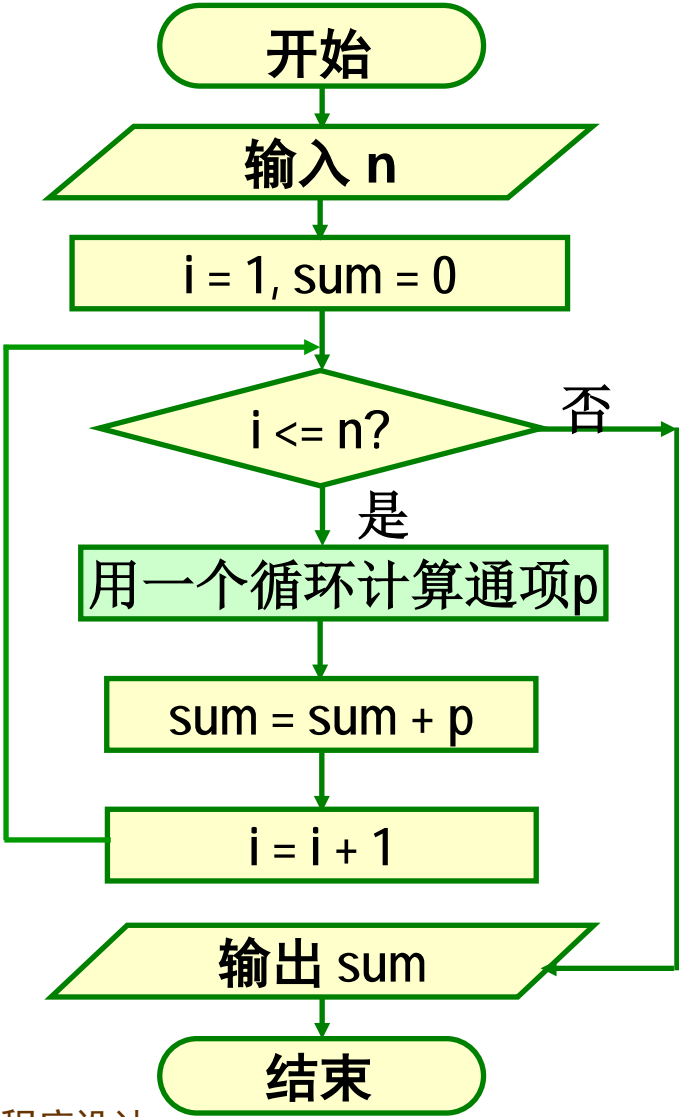
- 单独计算累加项p
- $\text{sum} = \text{sum} + p$

1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9

$$i! = (i-1)! * i$$

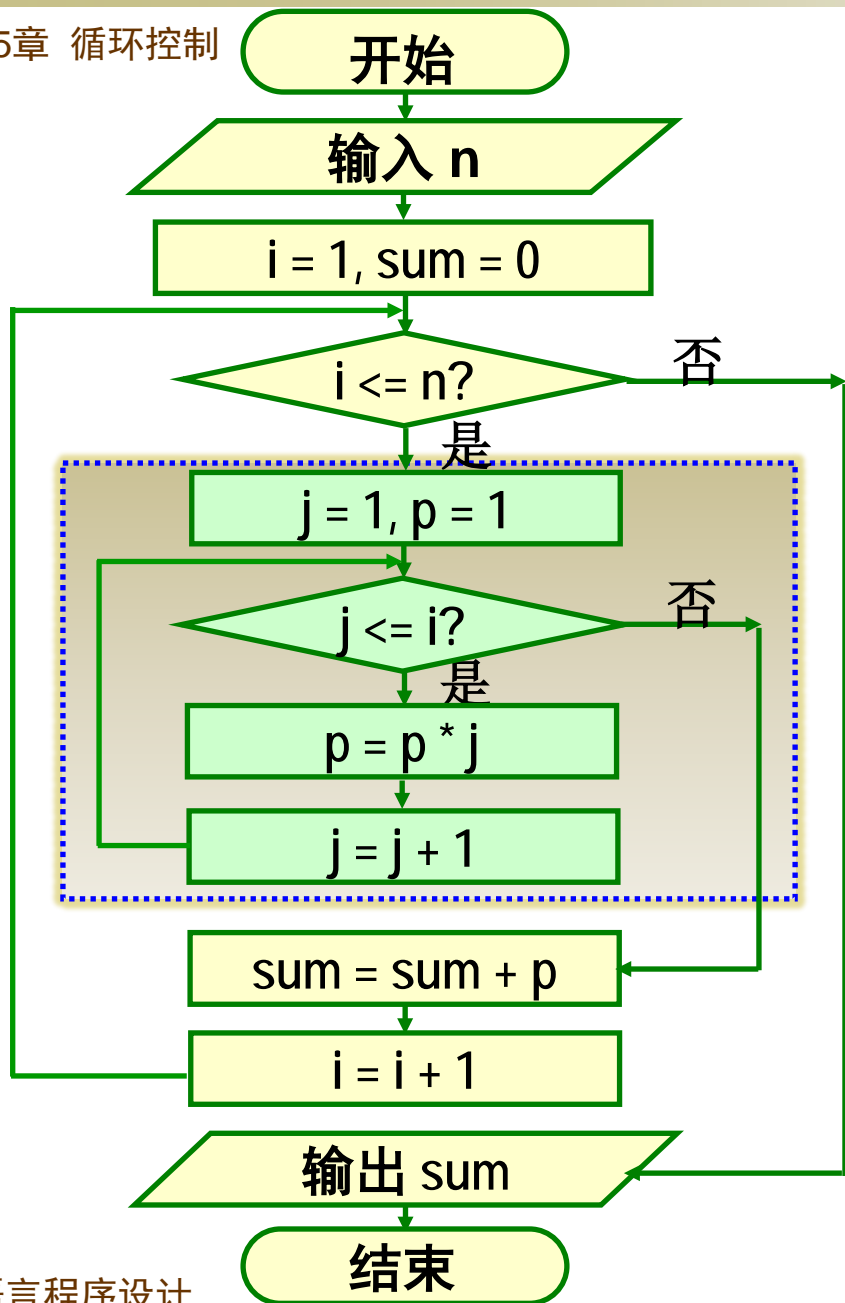
$$i! = 1 \times 2 \times 3 \times \dots \times i$$

计算并输出1! + 2! + 3! ... + n!



1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9

$i! = 1 \times 2 \times 3 \times \dots \times i$



```
#include <stdio.h>
int main()
```

```
{
```

```
int i, j, n;
```

```
long p, sum = 0;
```

```
printf("Input n:");
```

```
scanf("%d", &n);
```

```
for (i=1; i<=n; i++)
```

```
{
```

```
    p = 1;
```

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

```
    {
```

```
        p = p * j;
```

```
    }
```

```
    sum = sum + p;
```

```
}
```

```
printf("sum = %ld\n", sum);
```

```
return 0;
```

```
}
```

嵌套循环 (Nested Loop)

```
1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9
```

嵌套循环是如何执行的?



计算并输出1! + 2! + 3! ... + n!

i	j	内循环p (累加项)	外循环sum 旧值	外循环sum 新值
1	1	1(1!)	0	1
2	1	1		
	2	2(2!)	1	3
3	1	1		
	2	2		
	3	6(3!)	3	9
4	1	1		
	2	2		
	3	6		
	4	24(4!)	9	33
...				

```
#include <stdio.h>
int main()
{
    int i, j, n;
    long p, sum = 0;
    printf("Input n:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        p = 1;
        for (j=1; j<=i; j++)
        {
            p = p * j;
        }
        sum = sum + p;
    }
    printf("sum = %ld\n", sum);
    return 0;
}
```

嵌套循环的执行过程

1

1*2

1*2*3

1*2*3*4

1*2*3*4*5

1*2*3*4*5*6

1*2*3*4*5*6*7

1*2*3*4*5*6*7*8

1*2*3*4*5*6*7*8*9

sum = sum + 1!

sum = sum + 2!

sum = sum + 3!

sum = sum + 4!

嵌套循环的设计

- 实例——输出九九乘法表
- 由内到外
 - 先考虑每一行中的数据如何输出
 - 再考虑如何输出这样的多个行

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

输出九九乘法表

第1步考虑如何输出第m行的9个数

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

n

1	1*2	1*3	1*4	1*5	1*6	1*7	1*8	1*9
2	2*2	2*3	2*4	2*5	2*6	2*7	2*8	2*9
3	3*2	3*3	3*4	3*5	3*6	3*7	3*8	3*9
4	4*2	4*3	4*4	4*5	4*6	4*7	4*8	4*9
5	5*2	5*3	5*4	5*5	5*6	5*7	5*8	5*9
6	6*2	6*3	6*4	6*5	6*6	6*7	6*8	6*9
7	7*2	7*3	7*4	7*5	7*6	7*7	7*8	7*9
8	8*2	8*3	8*4	8*5	8*6	8*7	8*8	8*9
9	9*2	9*3	9*4	9*5	9*6	9*7	9*8	9*9

m

输出第m行的
9个数

开始

m = 1

n = 1

n <= 9?

否

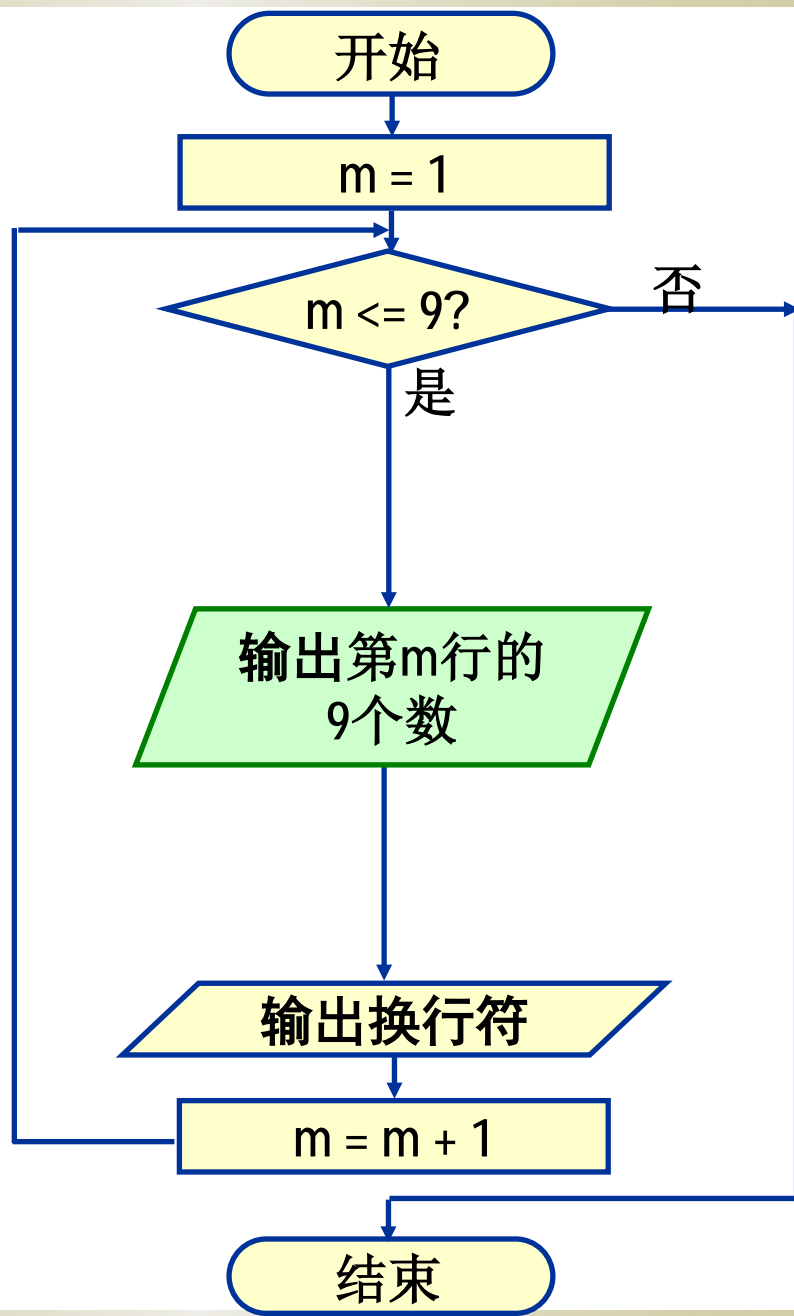
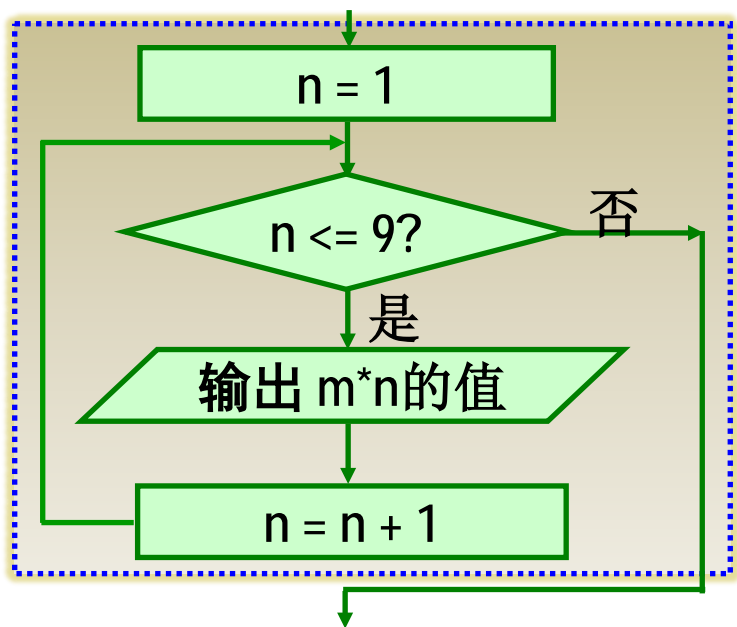
是

输出 m*n的值

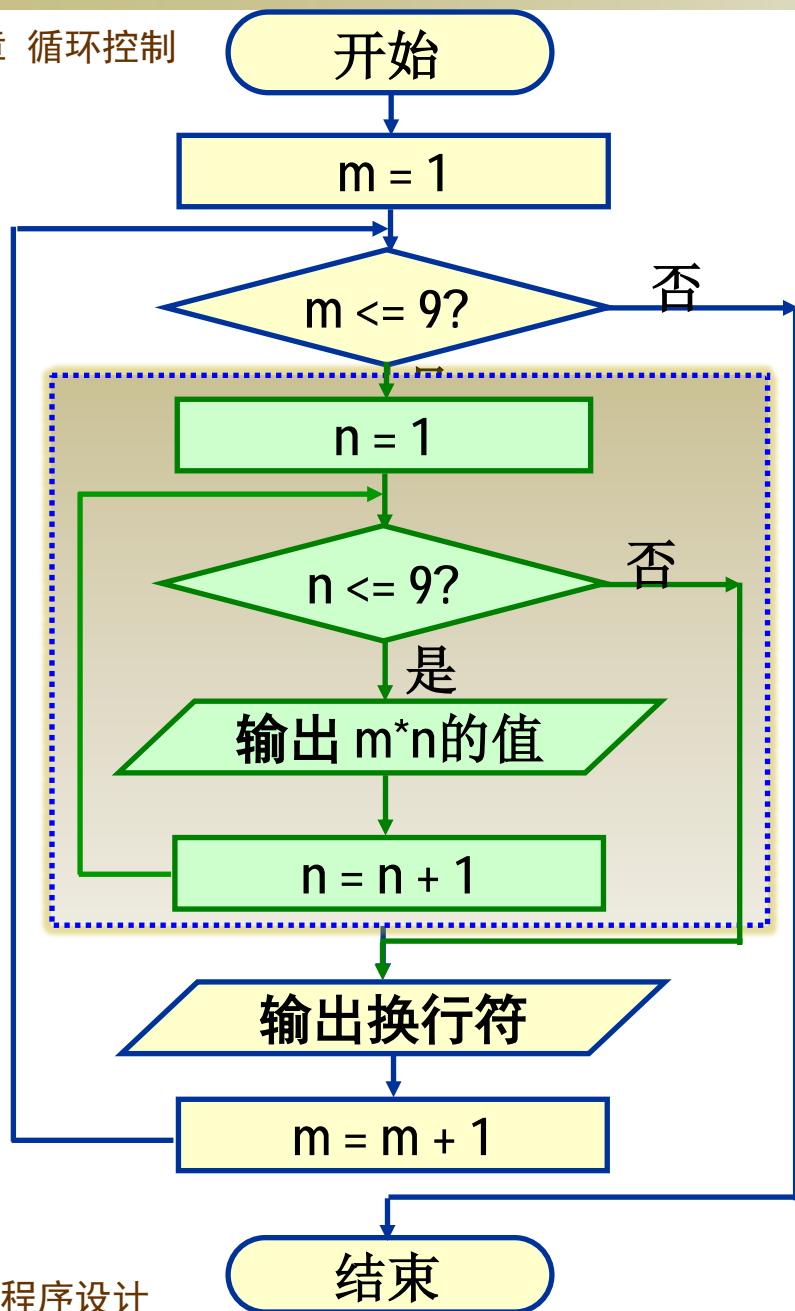
n = n + 1

结束

第2步考虑如何输出9个这样的行



输出九九乘法表



```
#include <stdio.h>
int main()
{
    int m, n;
    for (m=1; m<10; m++)
    {
        for (n=1; n<10; n++)
        {
            printf("%4d", m*n);
        }
        printf("\n");
    }
    return 0;
}
```

内层和外层循环
控制变量不能同名

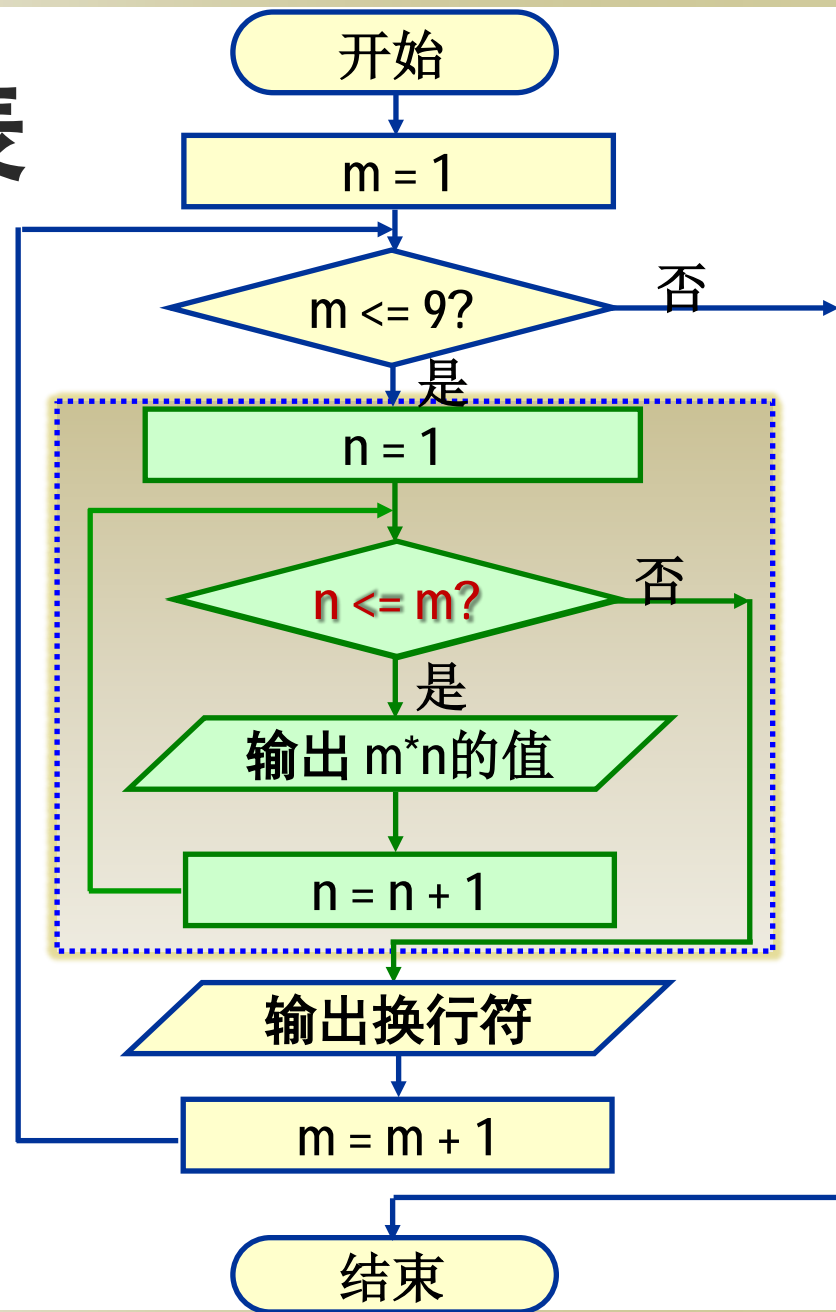
总循环次数 = 内循环
次数 × 外循环次数

输出九九乘法表

如何输出下三角的九九乘法表？

1									
2	4								
3	6	9							
4	8	12	16						
5	10	15	20	25					
6	12	18	24	30	36				
7	14	21	28	35	42	49			
8	16	24	32	40	48	56	64		
9	18	27	36	45	54	63	72	81	

1
1*2
1*2*3
1*2*3*4
1*2*3*4*5
1*2*3*4*5*6
1*2*3*4*5*6*7
1*2*3*4*5*6*7*8
1*2*3*4*5*6*7*8*9



讨论

如何输出上三角的九九乘法表？

```
1  2  3  4  5  6  7  8  9
   4  6  8 10 12 14 16 18
    9 12 15 18 21 24 27
   16 20 24 28 32 36
  25 30 35 40 45
 36 42 48 54
 49 56 63
 64 72
 81
```

