

# 第五章 流程控制

## 5.1 概述

C语言支持三种程序运行结构：**顺序结构**、**选择结构**、**循环结构**。

- 顺序结构：按照语句编写的顺序上到下逐句执。
- 选择结构：也叫分支结构，依据是否满足条件，有选择的执行相应功能。
- 循环结构：依据条件是否满足，重复多次执行某段代码。

## 5.2 选择结构

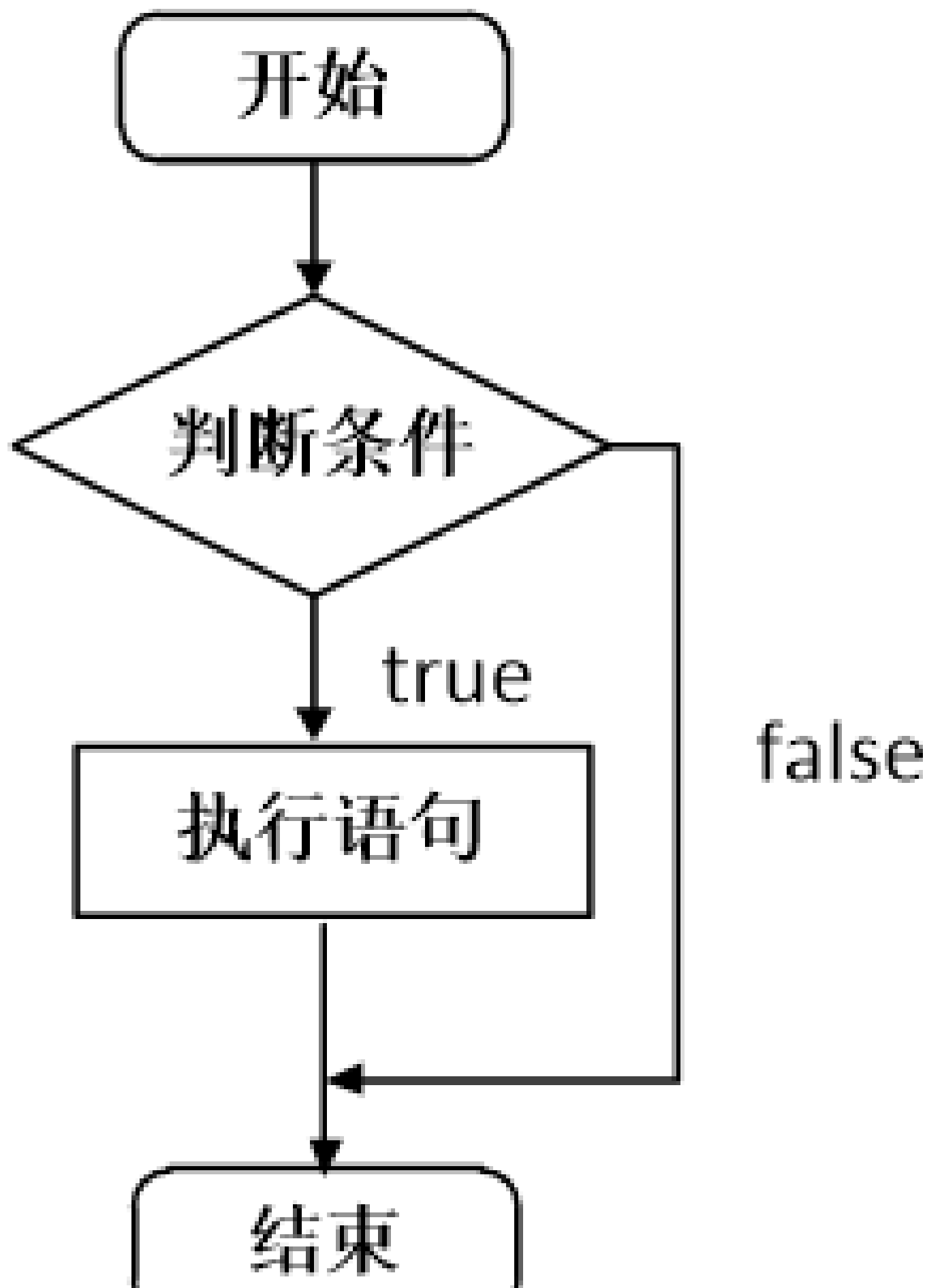
### 5.2.1 if语句

1、语法：

```
if (表达式)
{
    语句
}
```

2、执行逻辑

如果表达式的值为真，则执行其后的语句， 否则不执行该语句



### 3、示例代码：

```
#include <stdio.h>

int main()
{
    int a;
    scanf("Plz Input a Integer Number: ", &a);
    if (a > 10)
    {
        printf("a > 10 \n");
    }
    return 0;
}
```

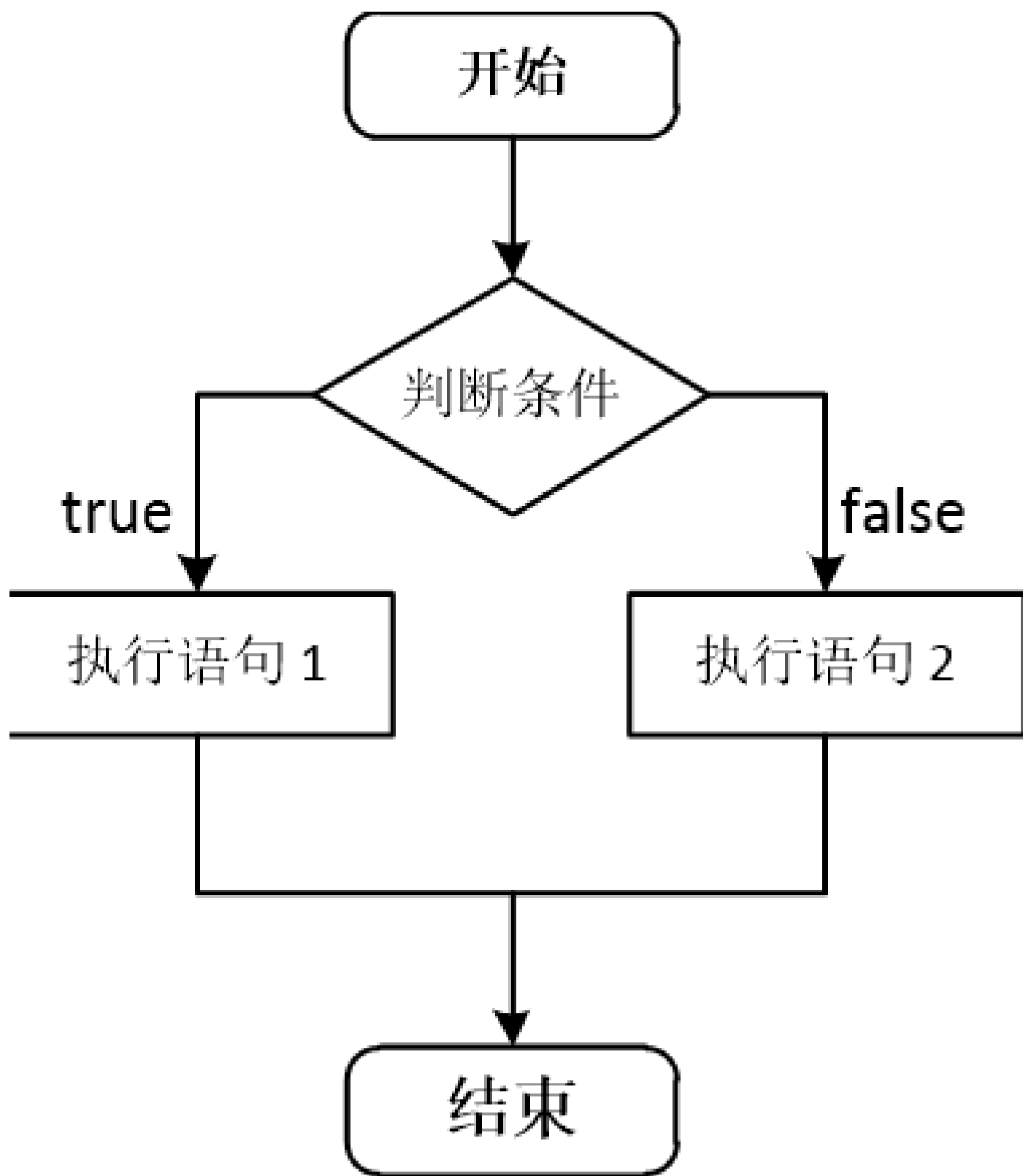
## 5.2.2 if...else语句

### 1、语法

```
if (表达式)
{
    语句1
}
else
{
    语句2
}
```

### 2、执行逻辑

如果表达式1成立（表达式1的结果为逻辑真），则执行语句1，否则执行语句2



### 3、示例代码

```
#include <stdio.h>
```

```
int main()  
{
```

```
int a;
printf("Plz Input a Integer Number: ");
scanf("%d", &a);
if (a > 10)
{
    printf("a > 10 \n");
}
else
{
    printf("a <= 10 \n");
}
return 0;
}
```

#### 4、练习

- 在键盘上输入三个整数，输出最大值
- 在键盘上输入一个年份，判断该年是否为闰年

### 5.2.3 if...else if ...else 语句

#### 1、语法

```
if (表达式1)
{
    语句1
}

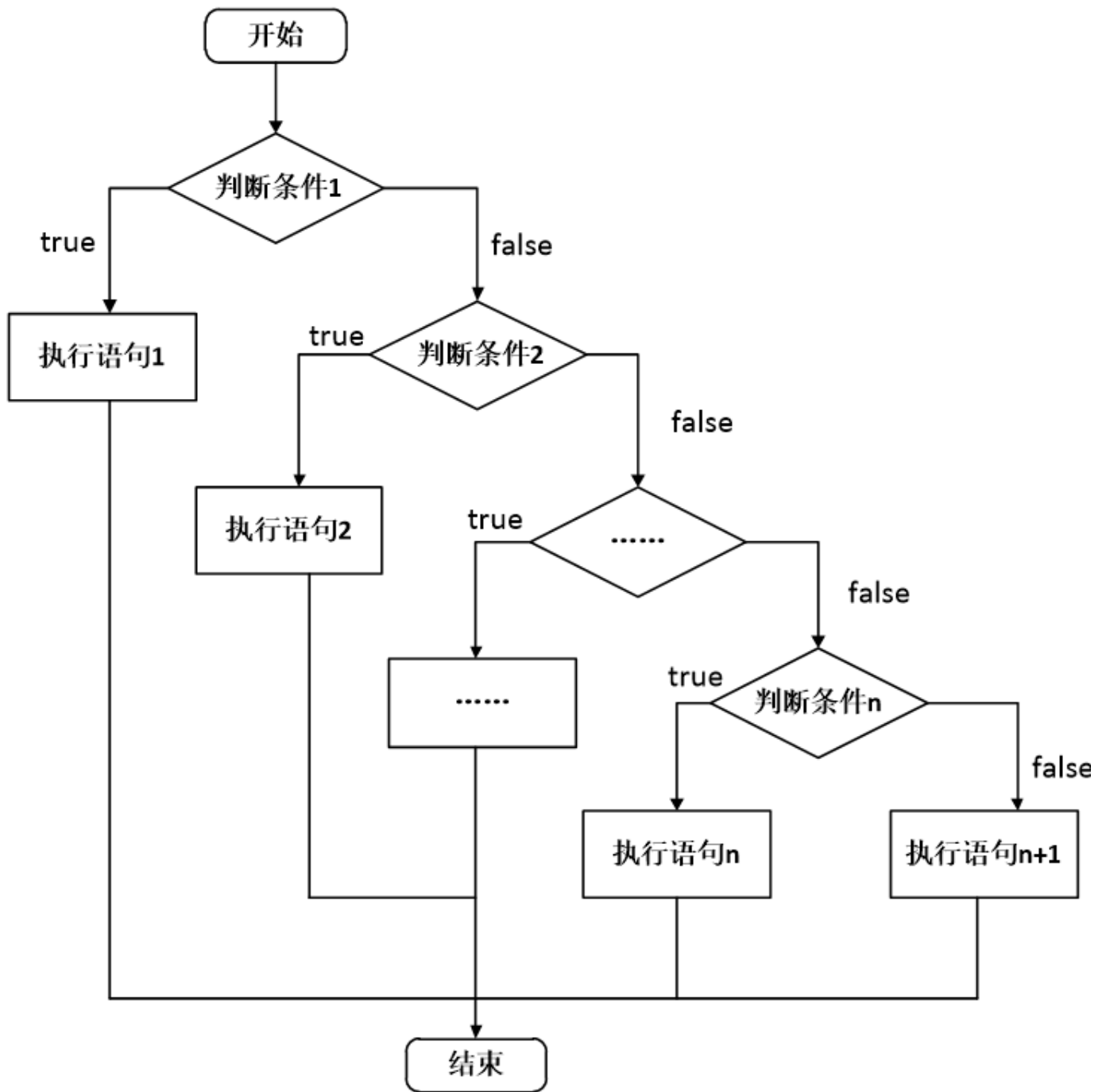
else if (表达式2)
{
    语句2
}

...

else if(表达式n)
{
    语句n-1
}
else
{
    语句n
}
```

#### 2、执行逻辑

从“表达式1”开始，从上往下判断，直到某个表达式的结果为逻辑真，则执行对应的语句，并且结束整个选择结构，如果所有的表达式结果为逻辑假，则执行else逻辑对应的“语句n”



### 3、示例代码

```
#include <stdio.h>

int main()
{
    unsigned int a;
    scanf("%u", &a);
```

```
if (a < 10)
{
    printf("个位\n");
}
else if (a < 100)
{
    printf("十位\n");
}
else if (a < 1000)
{
    printf("百位\n");
}
else
{
    printf("超过了4位数\n");
}

return 0;
}
```

## 5.2.4 switch语句

### 1、语法

```
switch (表达式)

{

case 常量表达式1: 语句1;break;

case 常量表达式2: 语句2;break;

...

case 常量表达式n-1: 语句n-1;break;

default                : 语句n

}
```

### 2、执行逻辑

计算"表达式"的值。并逐个与其后的常量表达式值相比较，当表达式的值与某个常量表达式的值相等时，即执行其后的语句，然后不再进行判断，继续执行后面所有case后的语句。如表达式的值与所有case后的常量表达式均不相同时，则执行default后的语句。

注意：每个语句后面都有break

### 3、示例代码

```
#include <stdio.h>

int main()
{
    int day;
    printf("plz Input weekday number: ");
    scanf("%d", &day);
    switch (day)
    {
        case 1:printf("星期一\n"); break;
        case 2:printf("星期二\n"); break;
        case 3:printf("星期三\n"); break;
        case 4:printf("星期四\n"); break;
        case 5:printf("星期五\n"); break;
        case 6:printf("星期六\n"); break;
        case 7:printf("星期日\n"); break;
        default:printf("I don't know \n");
    }
    return 0;
}
```

### 4、总结

- 1、switch结构的表达式必须为“常量表达式”
- 2、switch结构可以用if ... else if ... else 结构来实现

## 5.2.5 选择结构的嵌套

### 1、语法

```
if (表达式1)
{
    if (表达式2)
    {
        语句1
    }
    else
    {
        语句2
    }
}
```

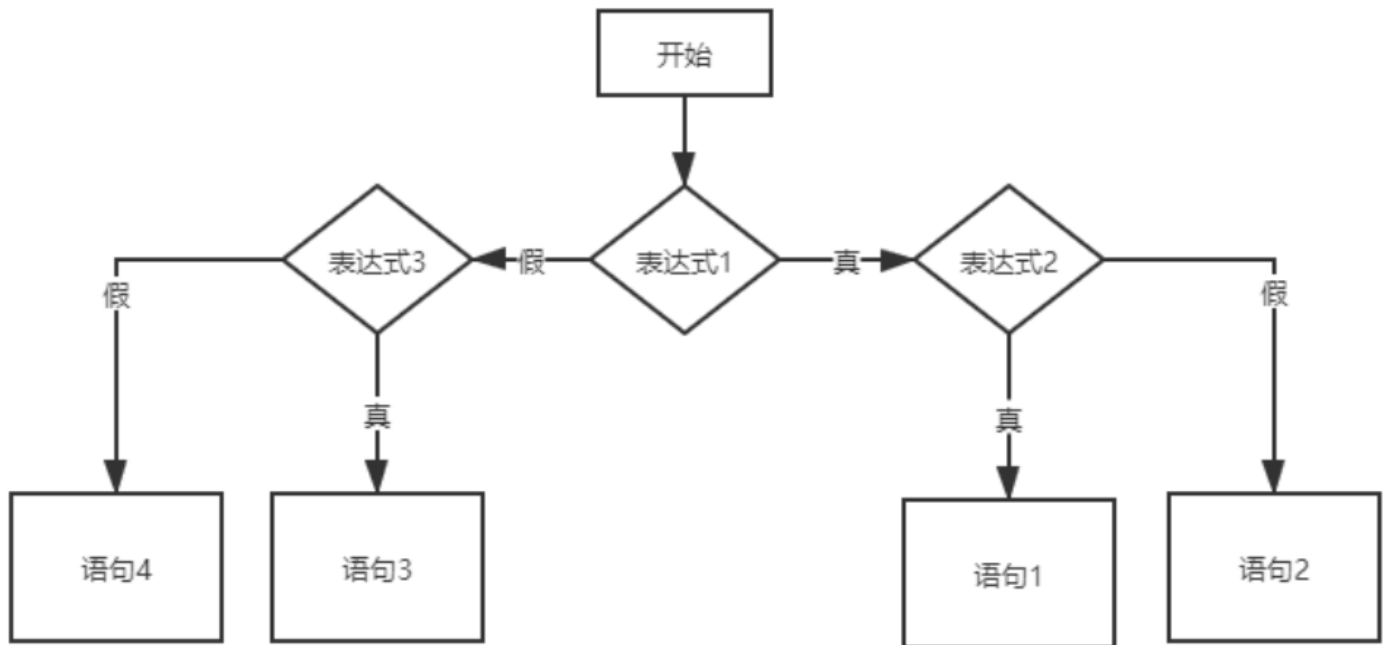


```
}  
else //else if ()  
{  
    if (表达式3)  
    {  
        语句3  
    }  
    else  
    {  
        语句4  
    }  
}
```

## 2、执行逻辑

如果表达式1结果为逻辑真，则判断表达式2的结果，如果结果为真则执行语句1，否则执行语句2

如果表达式1结果为逻辑假，则判断表达式3的结果，如果结果为真则执行语句3，否则执行语句4



## 3、举例

```
if (体育老师生病了)  
{  
    if (语文老师没课)  
    {  
        上语文课  
    }  
}
```

```
else
{
    上自习
}
else
{
    上体育课程
}
```

## 5.3 循环结构

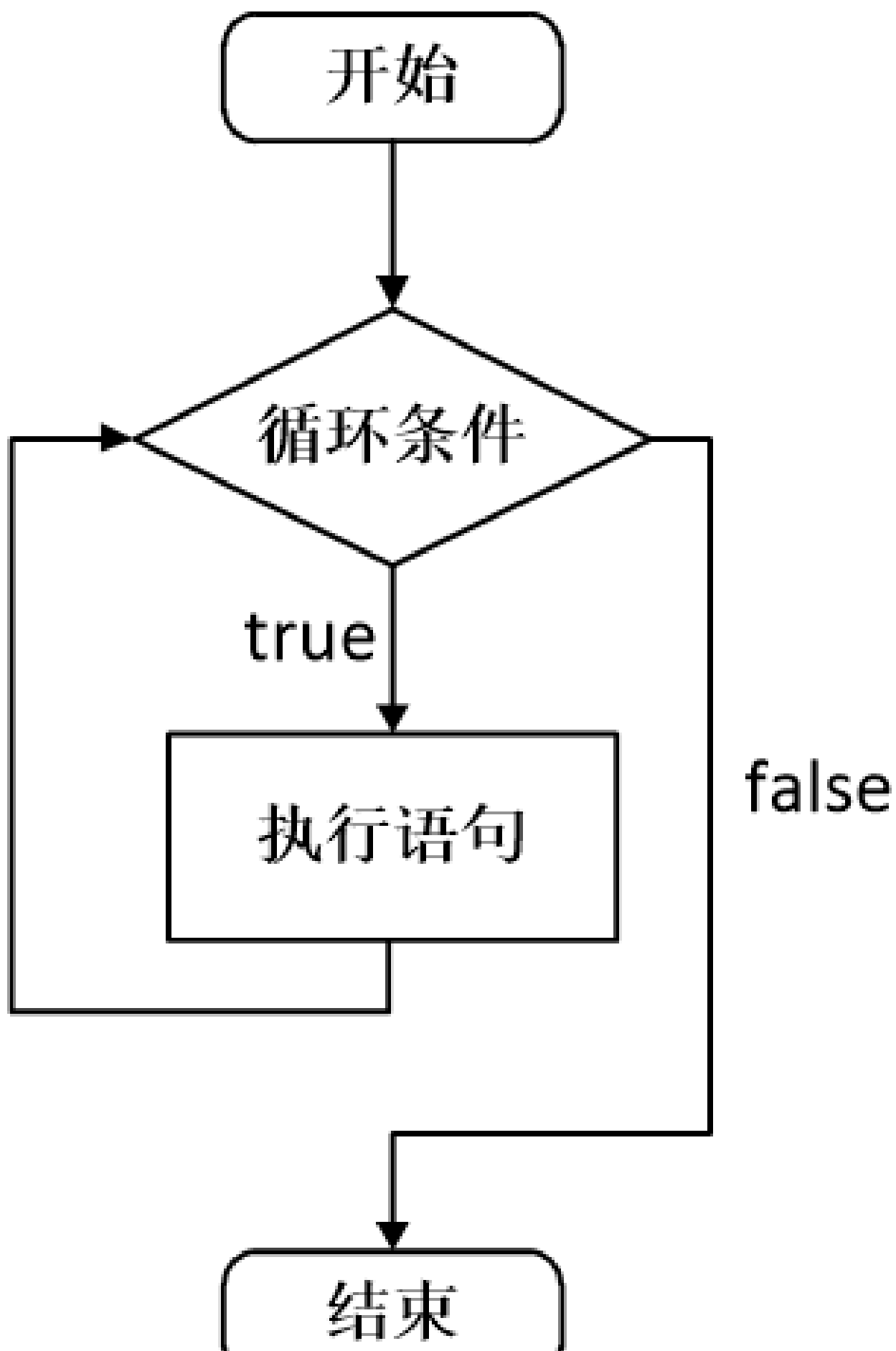
### 5.3.1 while语句

#### 1、语法

```
while (表达式)
{
    语句
}
```

#### 2、执行逻辑

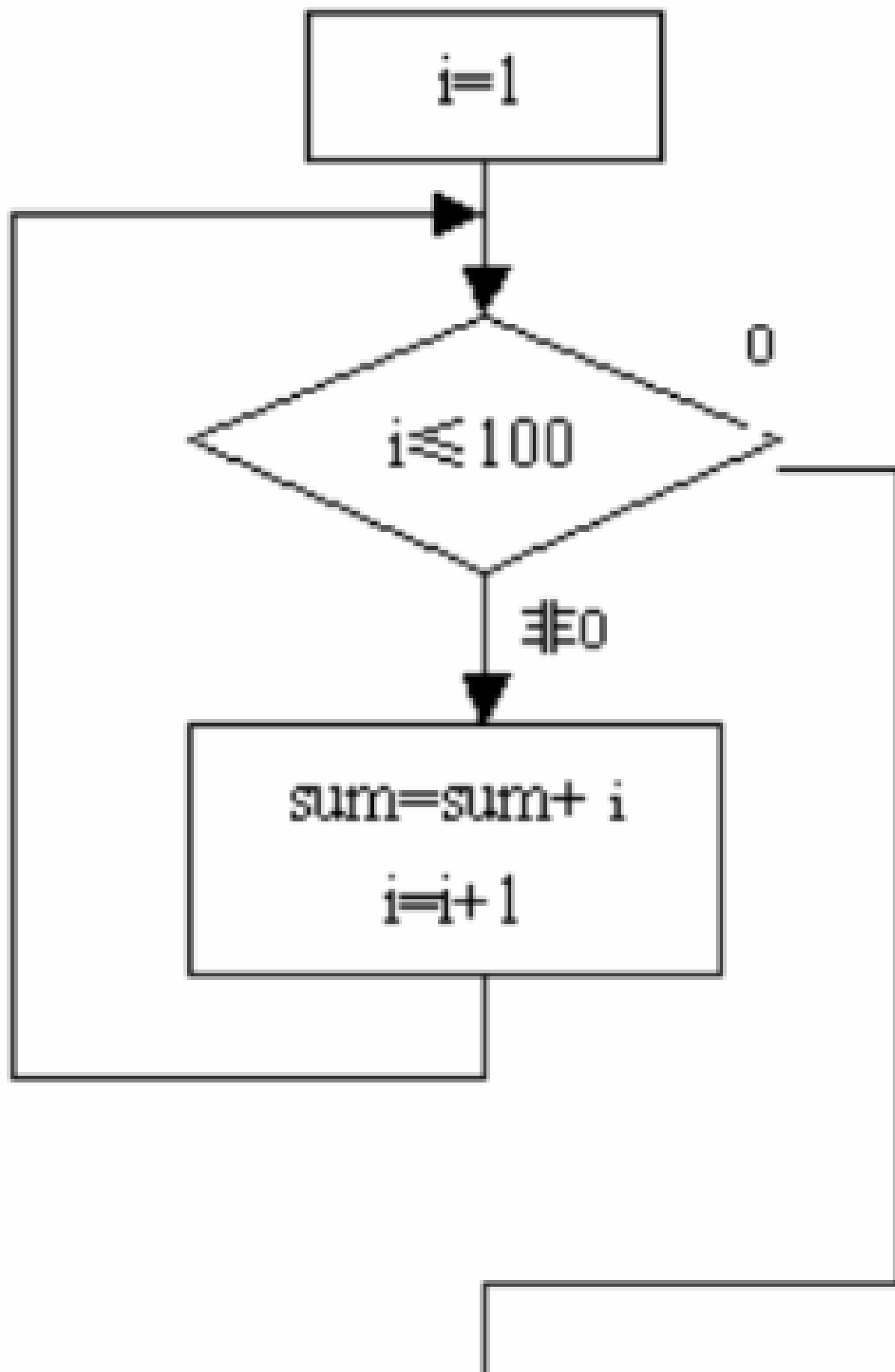
如果表达式的结果为逻辑假则执行循环体语句，直到表达式的结果为假



### 3、示例代码

```
#include <stdio.h>
int main()
{
    int x = 10;
    while (x != 0)
    {
        printf("%d\n", x);
        x--;
    }
    return 0;
}
```

### 4、练习：通过while循环计算1~100的和





```
#include <stdio.h>
int main()
{
    int sum = 0;
    int i = 1;
    while (i <= 100)
    {
        sum += i;
        i++;
    }
    return 0;
}
```

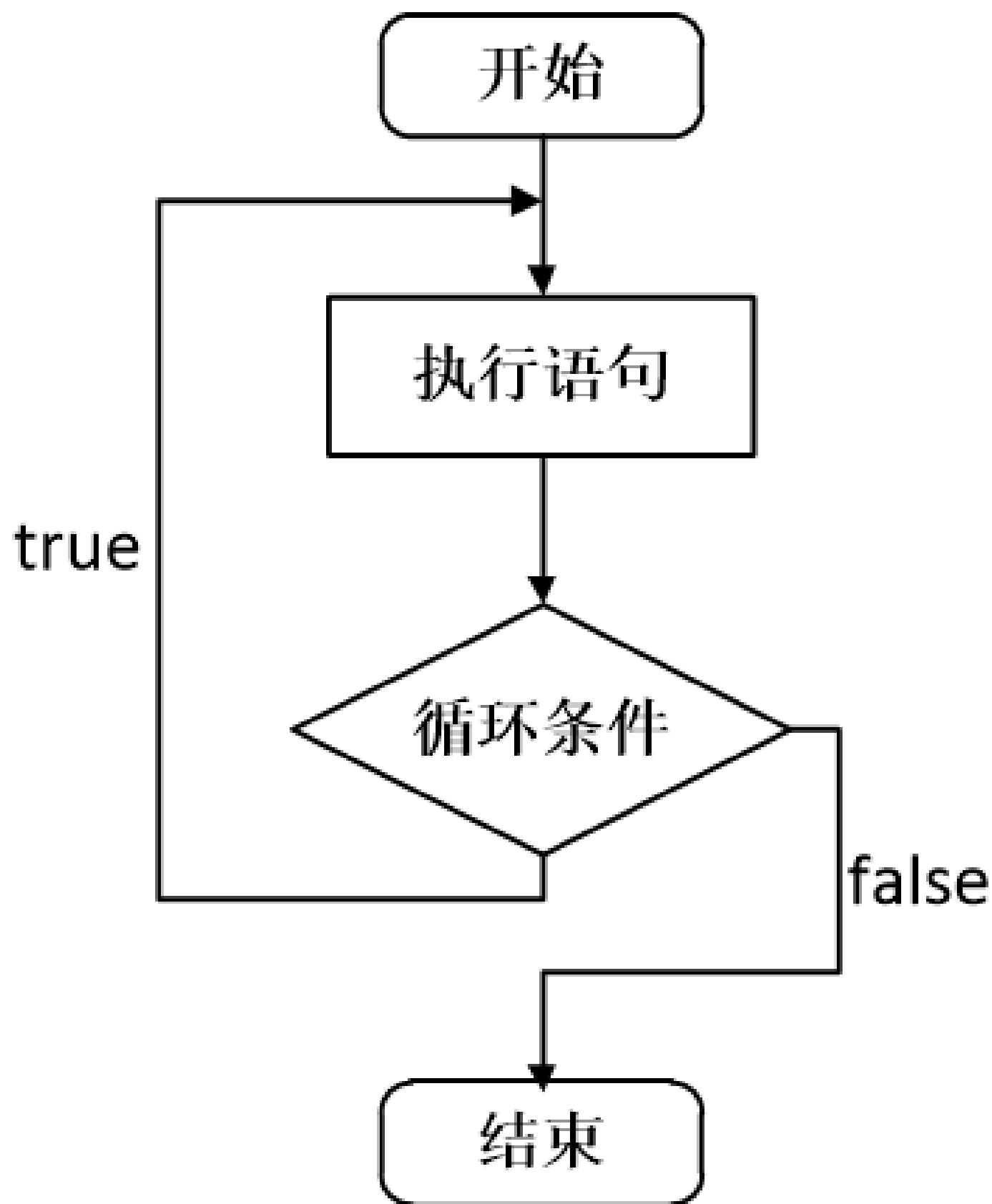
## 5.3.2 do~while 语句

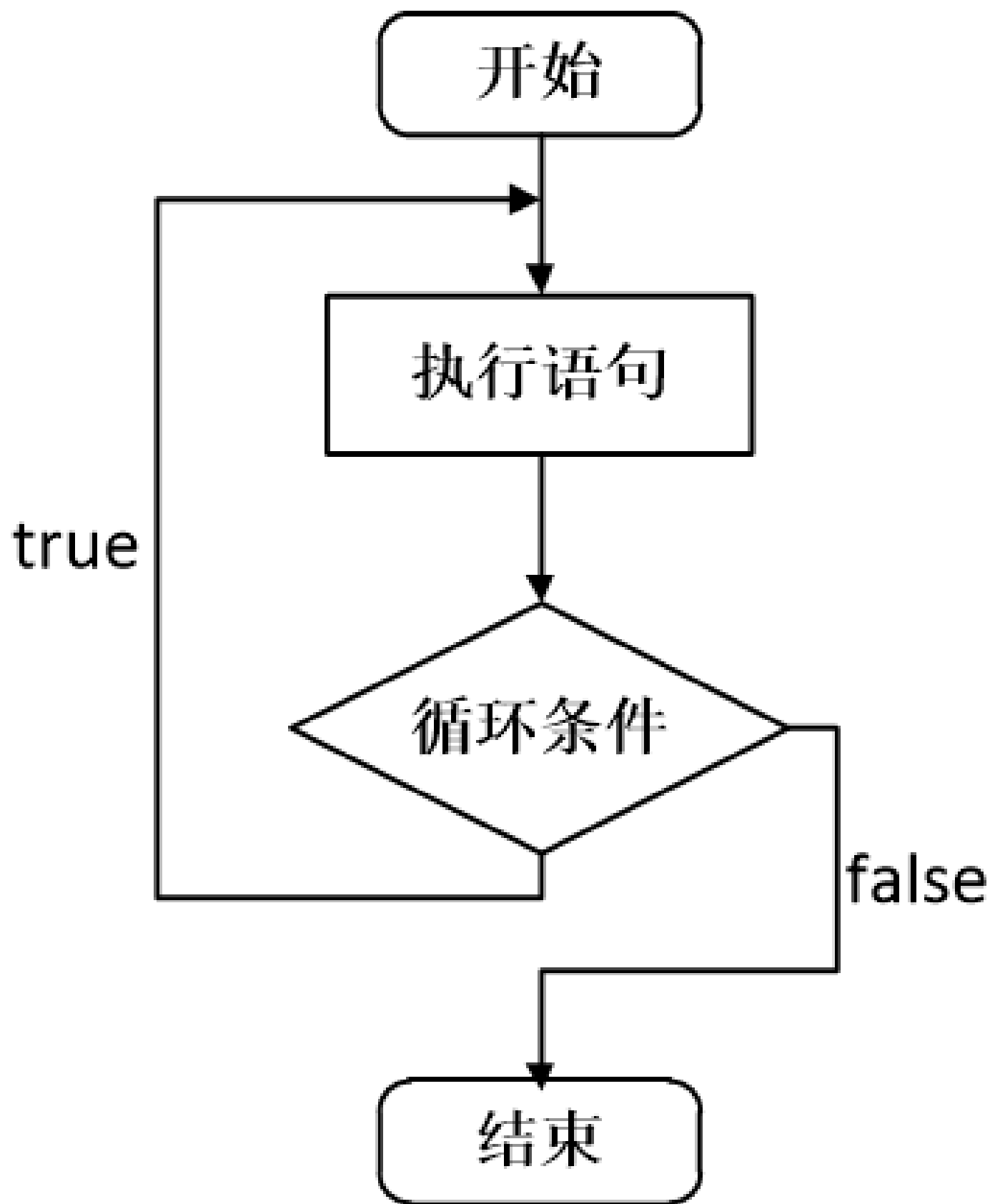
### 1、语法

```
do{
    语句
}while (表达式);
```

### 2、执行逻辑

这个循环与 while 循环的不同在于:它先执行循环中的语句,然后再判断表达式是否为真,如果为真则继续循环;如果为假,则终止循环。因此, do-while 循环至少要执行一次循环语句。其执行过程可用下图表示。





### 3、示例代码



```
#include <stdio.h>
int main()
{
    int sum = 0;
    int i = 1;
    do
    {
        sum += i;
        i++;
    }
    while (i <= 100);

    return 0;
}
```

4、总结：do~ while和while的区别在于：**do~while至少会执行一次循环体语句**

### 5.3.3 for语句

1、在C语言中，for语句使用最为灵活，它完全可以取代 while 语句。语法格式如下

```
for(表达式1; 表达式2; 表达式3)
{
    语句
}
```

2、执行逻辑

1. 先求解表达式1。
2. 求解表达式2，若其值为真(非0)，则执行for语句中指定的内嵌语句，然后执行下面第3)步;若其值为假(0)，则结束循环，转到第5)步。
3. 求解表达式3。
4. 转回上面第2)步继续执行。
5. 循环结束，执行for语句下面的一个语句。

求解表达式 1

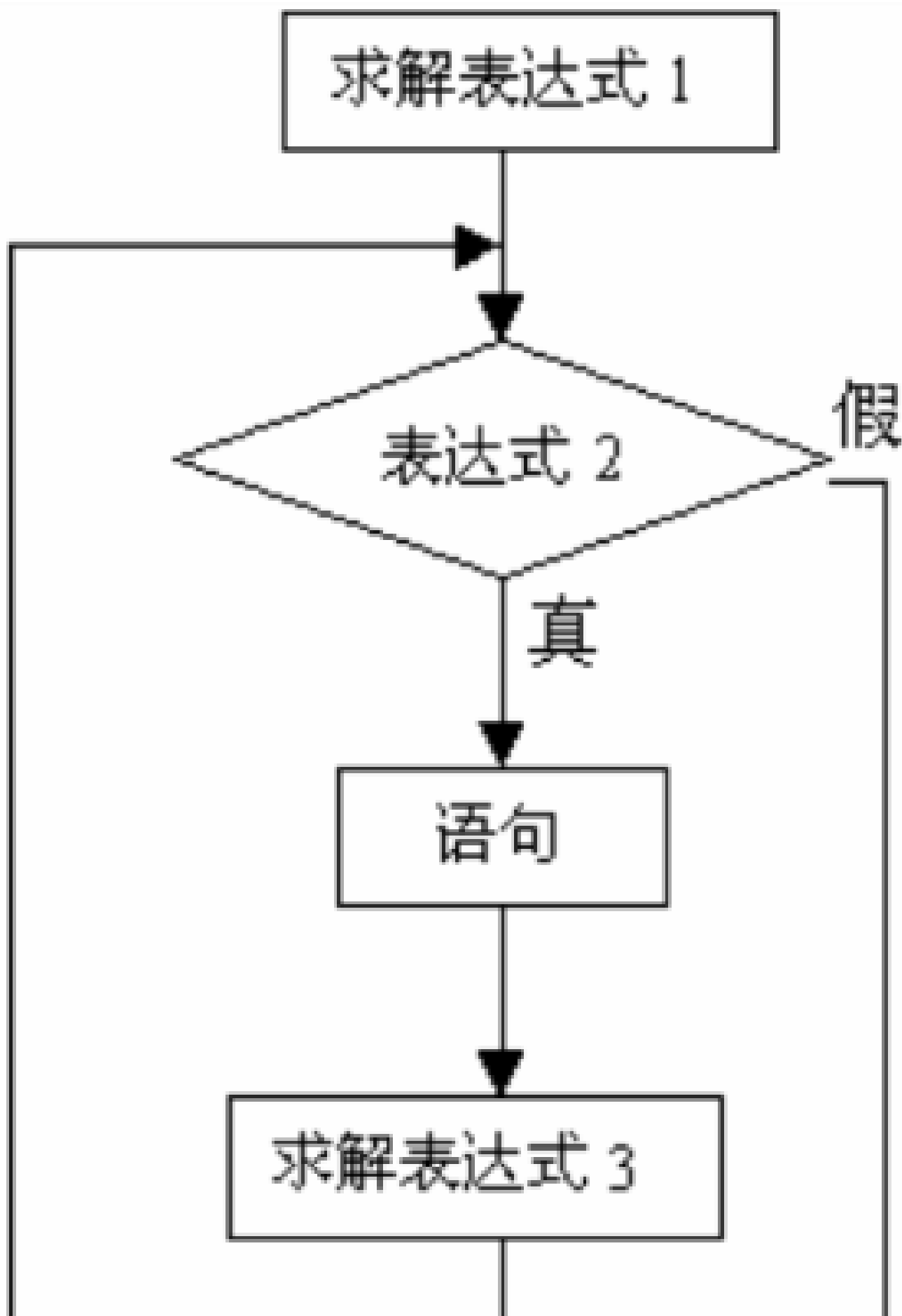
表达式 2

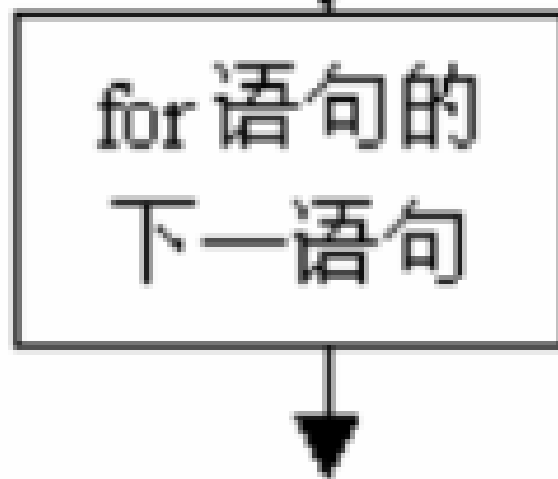
假

真

语句

求解表达式 3





### 3、示例代码

```
#include <stdio.h>
int main()
{
    int sum = 0;
    int i;
    for (i = 1; i <= 100; i++)
    {
        sum += i;
    }
    return 0;
}
```

### 4、实例讲解

求某个整数二进制表达形式中有多少个1

### 5、练习

- 求5的阶乘

- 输出100~999之间的所有的水仙花数（水仙花数是一个三位数满足如下条件：水仙花数 = 个位的立方+十位的立方+百位的立方）
- 在键盘上输入两个数字，分别存储到a和n中，自动计算表达式  $a + aa + aaa + \dots + a \dots a$  (n个a) 的结果，例如，在键盘上输入 3 和5，表达式为:  $3 + 33 + 333 + 3333 + 33333$

## 5.3.4 循环的嵌套

1、所谓的循环结构就是“循环逻辑中有循环逻辑”

2、示例代码

```
#include<stdio.h>
int main()
{
    int i,j;

    for(i=1;i<=9;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d*%d=%d\t",i,j,i*j);
        }
        printf("\n");
    }
    return 0;
}
```

3、实例讲解

求：2~100之间的所有的质数

## 5.4 流程控制

### 5.4.1 break语句

1、break在switch语句中的使用

- break用于switch中时,可使程序跳出switch,如果没有break语句,则会继续执行下一个case逻辑中的语句（不论case后面的表示式结果与switch后的表达式结果是否相等），直到遇到break才会跳出switch

```
switch(x)
{
```

```

case 1:printf("1\n");break;
case 2:printf("2\n");
case 3:printf("3\n");break;
case 4:printf("4\n");break;
}

```

- 上面的例子中因为case 2 逻辑后没有break，所以当x为2时，输出结果为 2 3
- 思考:如何使用switch语句实现在键盘上输入一个月份，然后输出该月份对应的天数呢？

## 2、break在循环语句中的使用

- 当break语句用于do-while、for、while循环语句中时,可使程序**终止循环**而执行循环后面的语句，通常break语句总是与if语句联在一起。即满足条件时便跳出循环。

```

int m, n;
for (m = 0; m < 10; m++)
{
    for (n = 0; n < 10; n++)
    {
        if (n == 3)
            break;
        printf("%d\n", n);
    }
}

```

## 5.4.2 continue语句

1、continue语句用于循环结构中，作用是结束本次循环，继续下次循环

```

int m, n;
for (m = 0; m < 10; m++)
{
    for (n = 0; n < 10; n++)
    {
        if (n == 3)
            continue;
        printf("%d\n", n);
    }
}

```

2、当执行到continue语句时，此次循环结构中continue语句后的其他所有语句都不会被执行到

## 5.5 企业笔试题

- 1、有一只青蛙，有一口井深  $h$  米，白天爬  $m$  米，晚上掉  $n$  米，假设这只青蛙从白天开始爬，多少天可以爬出来
- 2、在一个坐标系中有  $x$  坐标、 $y$  坐标，假如以坐标  $(400, 240)$  为圆心，请编程实现，打印出所有在以 50 为半径的圆中的所有的点的坐标
- 3、在键盘上输入一个数字，求除该数字的所有质因数（因数是质数，例如： $6 = 2 \times 3$ ， $20 = 2 \times 2 \times 5$ ）
- 4、假设某人有 100,000 现金. 每经过一次路口需要进行一次交费. 交费规则为当他现金大于 50,000 时每次需要交 5% 如果现金小于等于

50,000 时每次交 5,000. 请写一程序计算此人可以经过多少次这个路口

5、假设一个简单的 ATM 机的取款过程是这样的：

1) 首先提示用户输入密码 (password)，最多只能输入三次，超过 3 次则提示用户“密码错误，请取卡”结束交易。

2) 如果用户密码正确，再提示用户输入取款金额 (amount)

3) ATM 机只能输出 100 元的纸币，一次取钱数要求最低 100 元，最高 1000 元。

4) 若用户输入的金额符合上述要求，则打印输出用户取得钱数，最后提示用户“交易完成，请取卡”，否则提示假设用户密码是 111111，请编程实现。