



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

循环的终止、加快和嵌套

- ◆ 1、终止循环
- ◆ 2、加快循环
- ◆ 3、嵌套循环

3.5.4 break语句

- ▶ break语句
- ▶ break语句的作用是结束switch语句和循环语句的运行，转到后续语句，语法形式为：

```
break;
```

- ▶ break语句只能用在switch语句和循环语句（while、do、for）中，不得单独使用，而且break语句只结束包含它的switch语句和循环语句，不会将所有嵌套语句结束。

3.5.4 break语句

- ▶ 显然，在循环结构中使用break语句的目的就是提前结束循环。
- ▶ C语言的三个循环语句如果循环条件恒为真时，循环会无终止地执行下去，如果在循环体中执行到break语句，循环就会结束，此时的循环就不是死循环。

3.5.4 break语句

- ▶ 因此，循环语句结束有两个手段：
- ▶ 一是循环条件
- ▶ 二是应用break语句。

3.5.4 break语句

- ▶ 由于循环体中使用break语句通常附带条件，例如：

```
if (m%i==0) break;
```

- ▶ 仍可以将break的应用理解为循环三要素的循环条件。

3.5.4 break语句



【例3.10】

判断一个数 m 是否素数，如果是输出Yes，否则输出No。

3.5.4 break语句



例题分析

所谓素数是指除了1和自己外，不能被其他数整除的数，例如17。判断方法是对2到 $m-1$ 的数逐个检查，如果 m 不能被其中任何一个数整除，那么 m 就是素数。

实际编程时，前述方法需要对所有数检查一遍。而利用反逻辑，即只要有一个数能被 m 整除，就不用再检查（ m 肯定不是素数）。

即使如此，如何在循环检查结束时就知道 m 是否为素数呢？这里可以测试循环是如何结束的。

3.5.4 break语句

例3.10

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,m;
5     scanf("%d",&m);
6     //从2到m-1之间逐一检查是否被m整除
7     for (i=2 ; i<=m-1 ; i++)
8         if (m % i==0) break; //如果整除则结束检查
9     if (i==m) printf("Yes\n"); //根据循环结束位置判断是否素数
10    else printf("No\n");
11    return 0;
12 }
```

3.5.4 break语句

例3.10

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,m;
5     scanf("%d",&m);
6     //从2到m-1之间逐一检查是否被m整除
7     for (i=2 ; i<=m-1 ; i++)
8         if (m % i==0) break; //如果整除则结束检查
9     if (i==m) printf("Yes\n"); //根据循环结束位置判断是否素数
10    else printf("No\n");
11    return 0;
12 }
```

3.5.4 break语句

例3.10

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,m;
5     scanf("%d",&m);
6     //从2到m-1之间逐一检查是否被m整除
7     for (i=2 ; i<=m-1 ; i++)
8         if (m % i==0) break; //如果整除则结束检查
9     if (i==m) printf("Yes\n"); //根据循环结束位置判断是否素数
10    else printf("No\n");
11    return 0;
12 }
```

3.5.4 break语句

例3.10

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,m;
5     scanf("%d",&m);
6     //从2到m-1之间逐一检查是否被m整除
7     for (i=2 ; i<=m-1 ; i++)
8         if (m % i==0) break; //如果整除则结束检查
9     if (i==m) printf("Yes\n"); //根据循环结束位置判断是否素数
10    else printf("No\n");
11    return 0;
12 }
```

3.5.5 continue语句

- ▶ continue语句
- ▶ continue语句的作用是在循环体中结束本次循环，直接进入下一次循环，语句形式为：

```
continue;
```

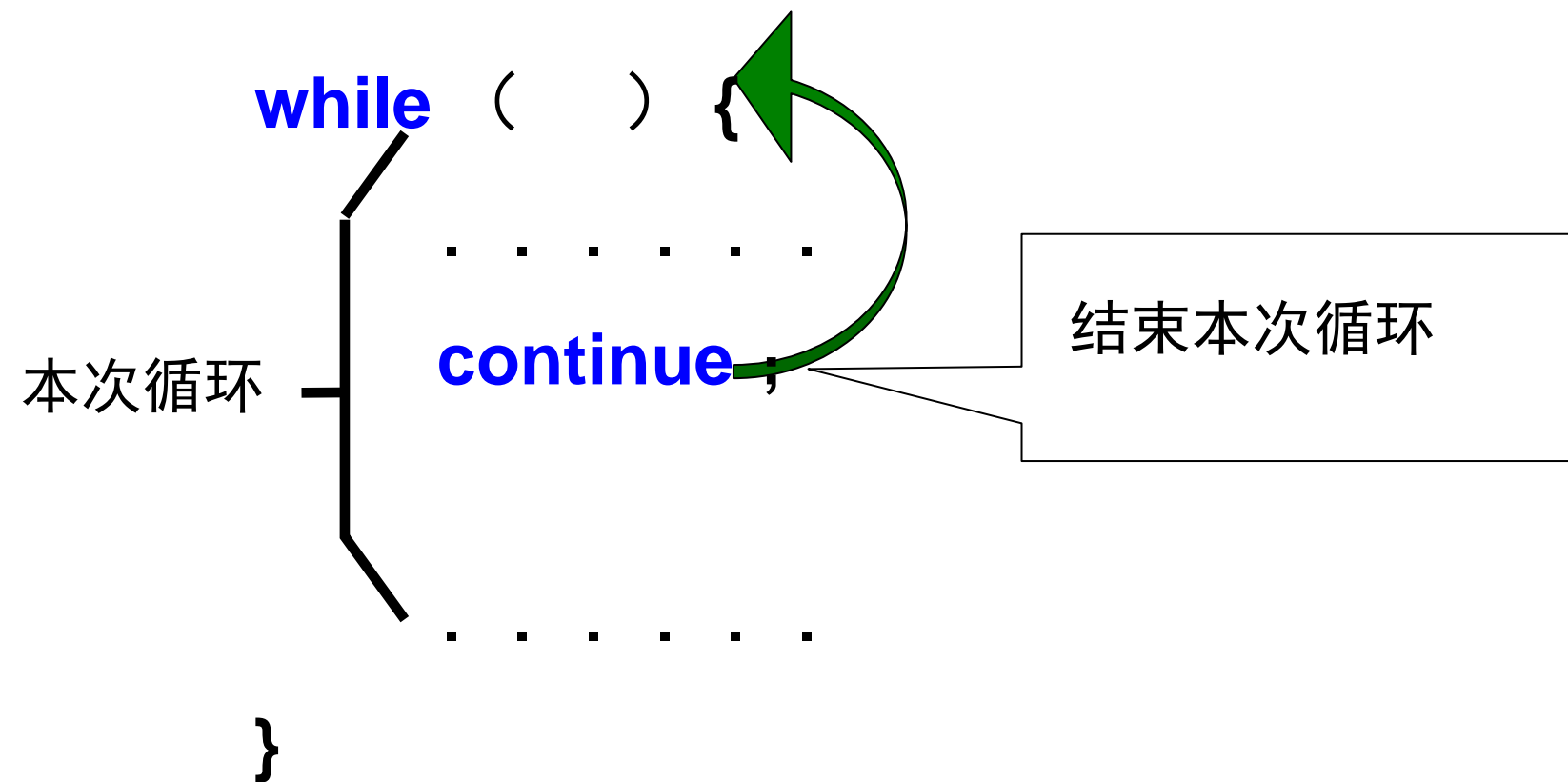
- ▶ continue语句只能用在循环语句（while、do、for）中，不能单独使用，而且continue语句只对包含它的循环语句起作用。

3.5.5 continue语句

- ▶ 在while语句和do语句循环体中执行continue语句，程序会转到“表达式”继续运行，在for语句循环体中执行continue语句，程序会转到“表达式3”继续运行，循环体中余下的语句被跳过了。所以continue语句的实际效果就是将一次循环结束，开始新的一次循环。

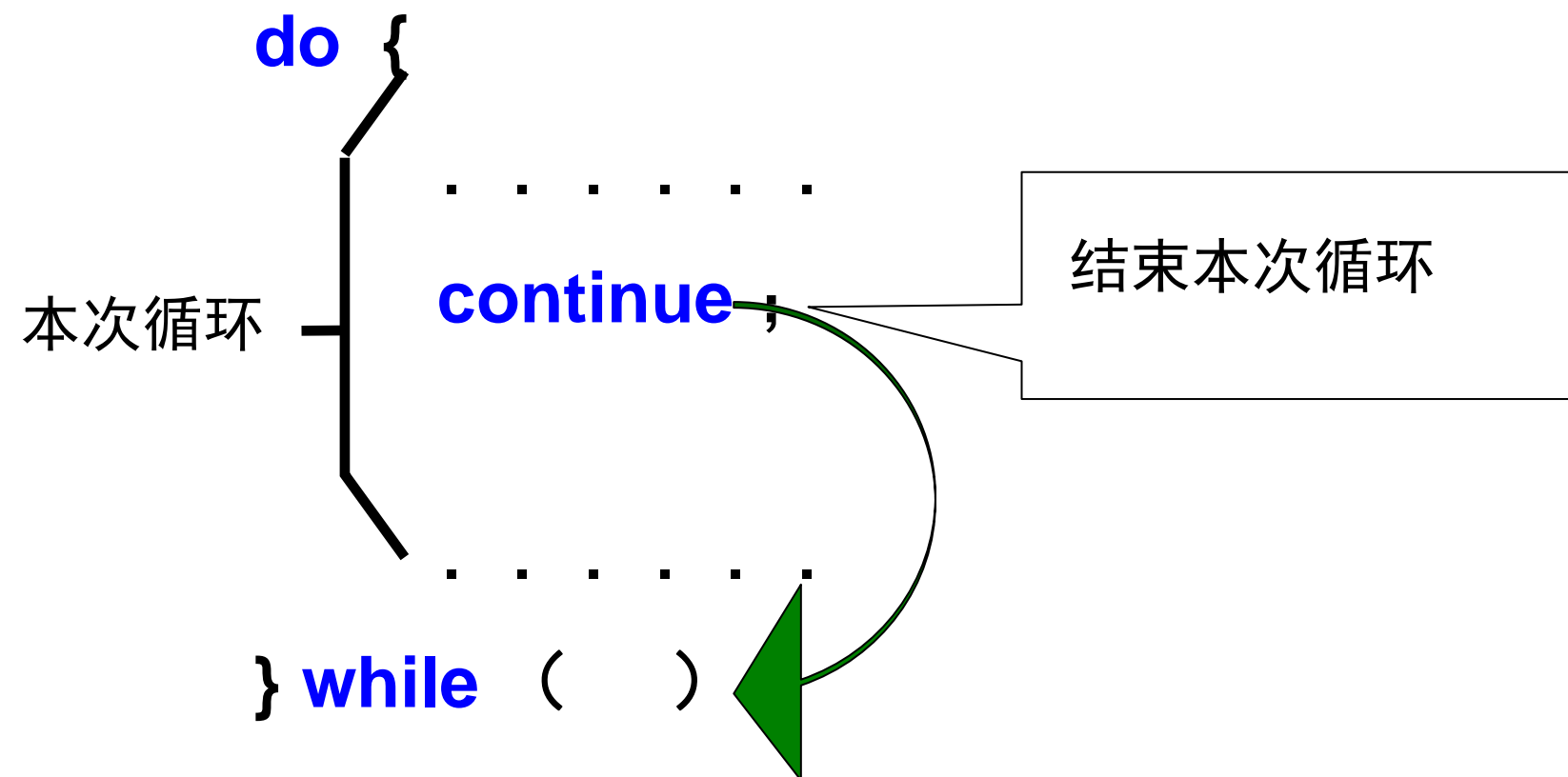
3.5.5 continue语句

► while 语句中的continue图示：



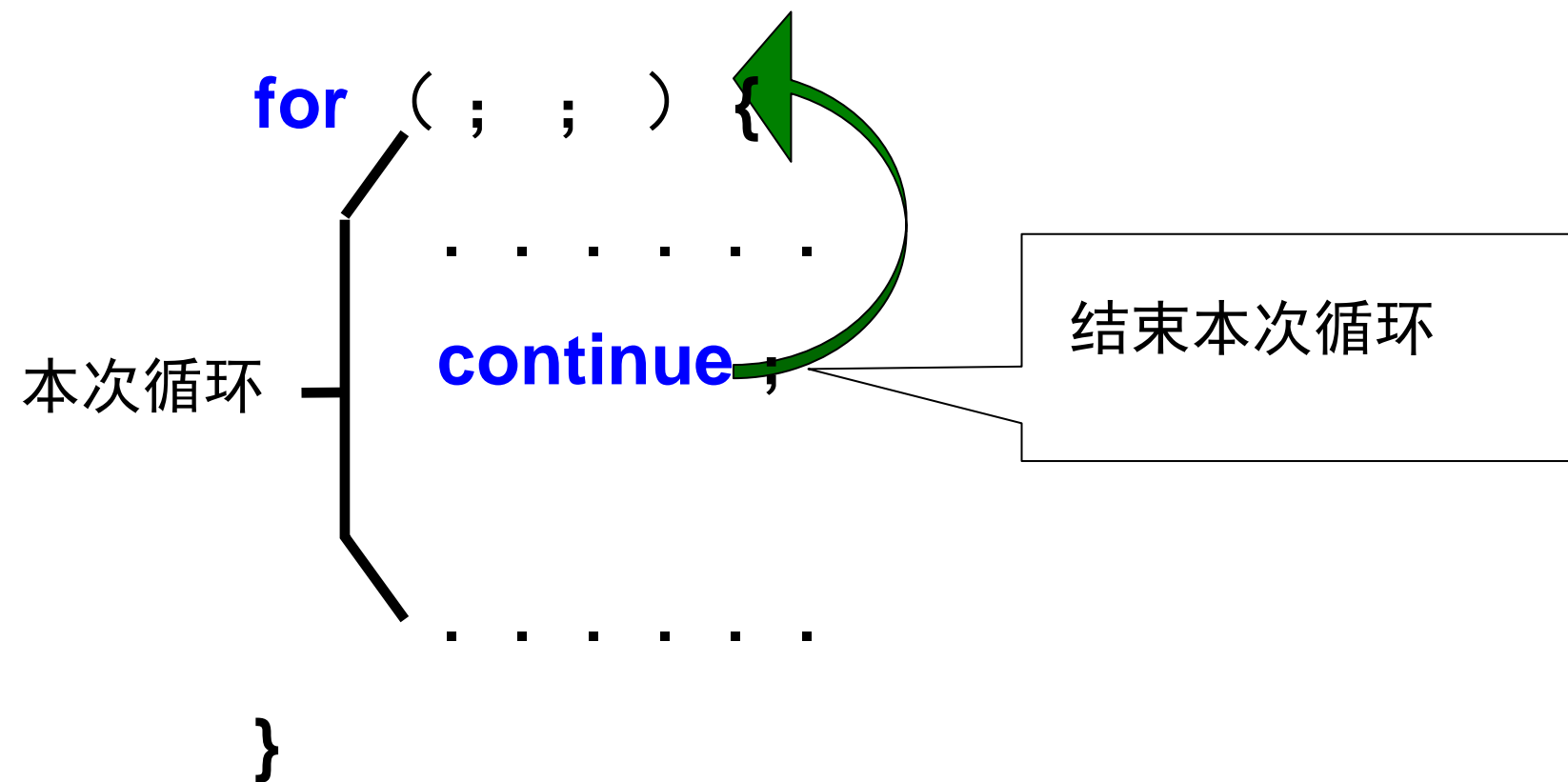
3.5.5 continue语句

► do-while 语句中的continue图示：



3.5.5 continue语句

► for 语句中的continue图示：



3.5.5 continue语句

▶ 比较下面两段程序。

▶ ①：

```
for (n=1, sum=0 ; n<=100 ; n++) {  
    if (n%2==0) break;  
    sum=sum+n;  
}
```

▶ 当if语句条件满足时（n为2），执行break，循环结束，故sum结果为0+1。

3.5.5 continue语句

► ②:

```
for (n=1, sum=0 ; n<=100 ; n++) {  
    if (n%2==0) continue;  
    sum=sum+n;  
}
```

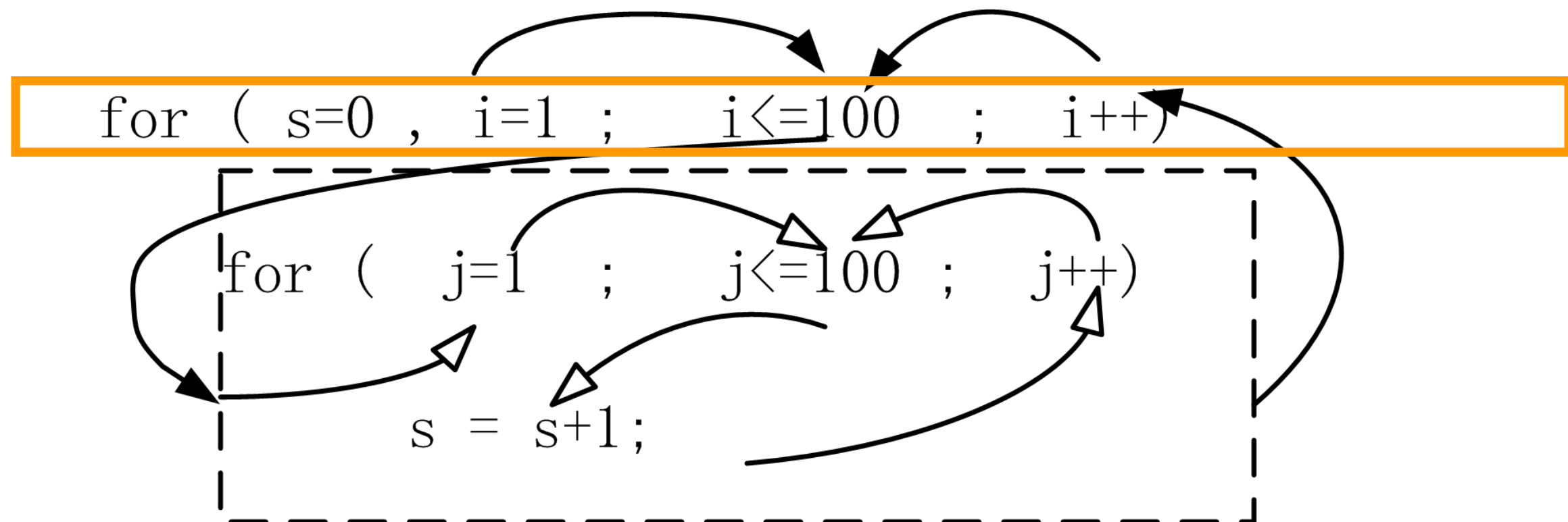
- 当if语句条件满足时（n为偶数），执行continue，则后面的累加语句被跳过，转到n++继续新的循环，故sum结果为0+1+3+5+...+99。

3.5.6 循环结构的嵌套

- ▶ 循环体可以是任意的控制语句，如果一个循环体内包含又一个循环语句时，就构成了循环结构的嵌套。C语言的循环语句（while、do、for）可以互相嵌套，循环嵌套的层数没有限制，可以形成多重循环。
- ▶ 多重循环的使用，进一步增加程序流程反复执行的次数，程序的循环能力更强。

3.5.6 循环结构的嵌套

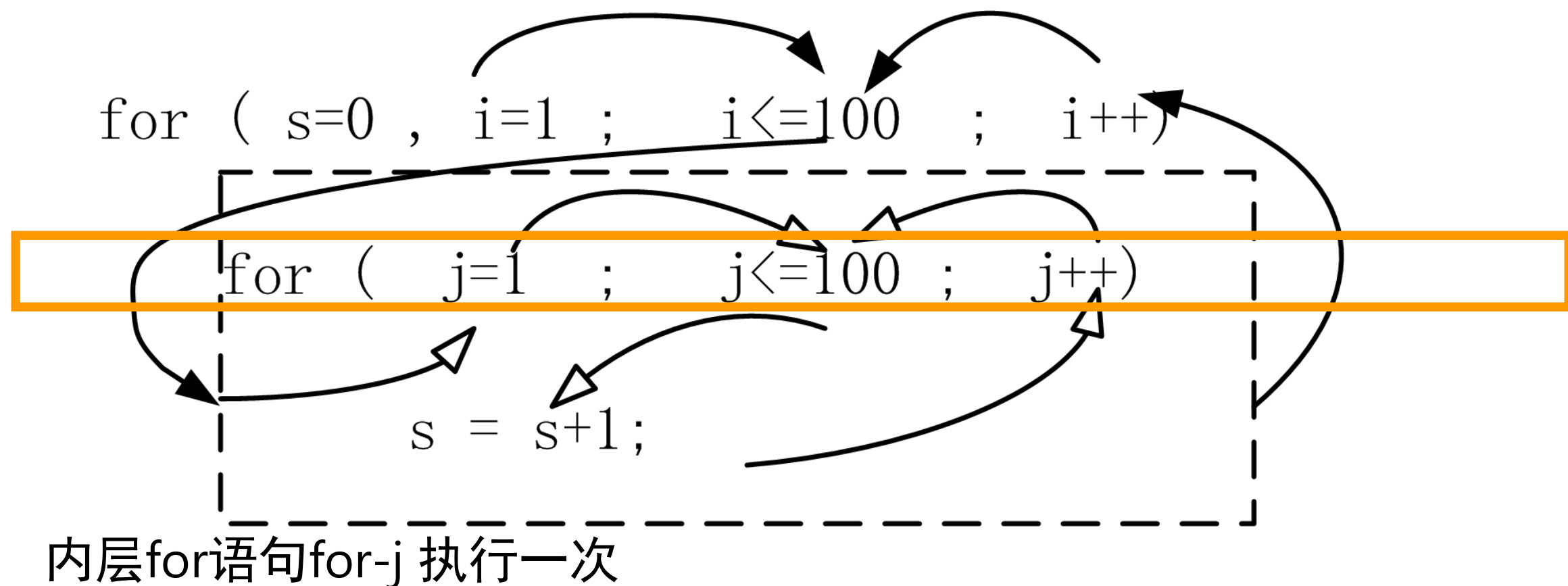
图3.12 双重循环执行流程



- (1) 首先执行外层for语句循环for-i, i=1
- (2) 当i<100条件为真

3.5.6 循环结构的嵌套

图3.12 双重循环执行流程

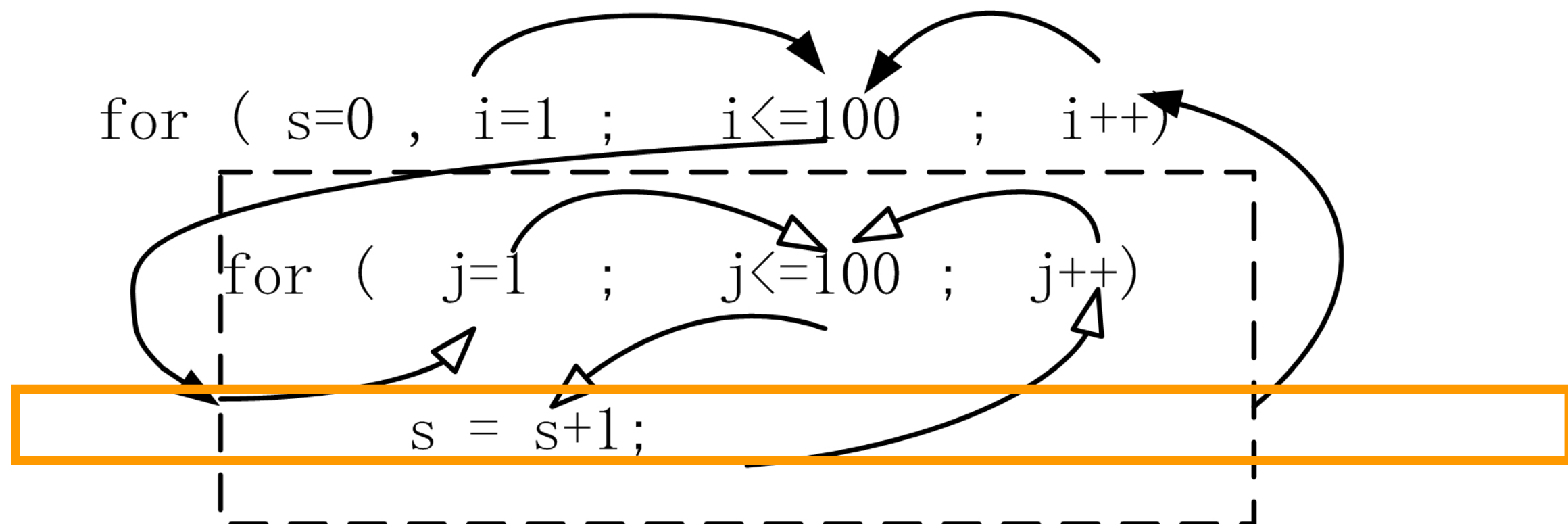


j从1执行到100，故for-j的循环体s=s+1执行100次

退出内层for语句

3.5.6 循环结构的嵌套

图3.12 双重循环执行流程



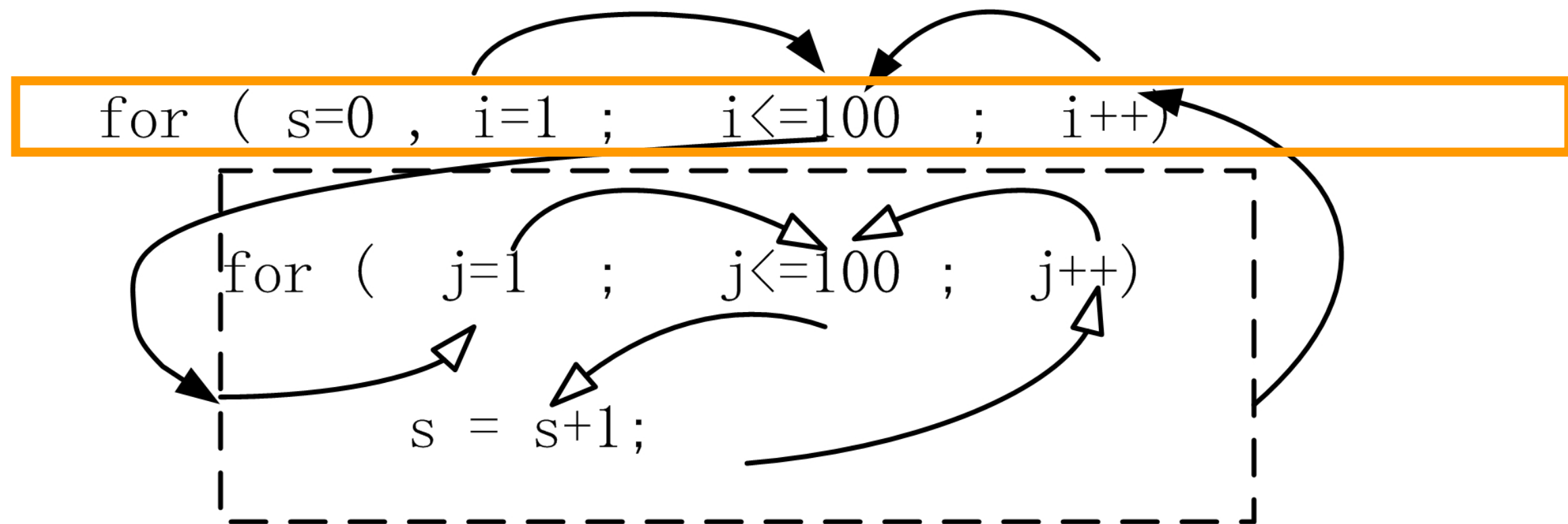
内层for语句for-j 执行一次

j从1执行到100，故for-j的循环体s=s+1执行100次

退出内层for语句

3.5.6 循环结构的嵌套

图3.12 双重循环执行流程



(3) 回到外层for语句循环的i++, 重复 (2)

3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

```
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

3.5.6 循环结构的嵌套

例3.53

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j;
5     for (i=1; i<=10; i++) { //控制行
6         for (j=1; j<=10; j++) //每行固定输出10个*
7             printf("*");
8         printf("\n"); //每行末尾输出1个换行
9     }
10    return 0;
11 }
```

3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

3.5.6 循环结构的嵌套

例3.54

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j;
5     for (i=1; i<=10; i++) { //控制行
6         for (j=1; j<=i; j++) //控制每行的*的个数：与行相关
7             printf("*");
8         printf("\n"); //每行末尾输出1个换行
9     }
10    return 0;
11 }
```

3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

```
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
```

3.5.6 循环结构的嵌套

例3.55

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j;
5     for (i=1; i<=10; i++) { //控制行
6         for (j=1; j<=10-i; j++) //每行先输出若干空格：与行相关
7             printf(" ");
8         for (j=1; j<=i; j++) //控制每行的*的个数：与行相关
9             printf("*");
10        printf("\n"); //每行末尾输出1个换行
11    }
12    return 0;
13 }
```

3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

```

      *
    ***
  *****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

3.5.6 循环结构的嵌套

例3.56

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j;
5     for (i=1; i<=10; i++) { //控制行
6         for (j=1; j<=10-i; j++) //每行先输出若干空格：与行相关
7             printf(" ");
8         for (j=1; j<=2*i-1; j++) //控制每行的*的个数：与行相关
9             printf("*");
10        printf("\n"); //每行末尾输出1个换行
11    }
12    return 0;
13 }
```


3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

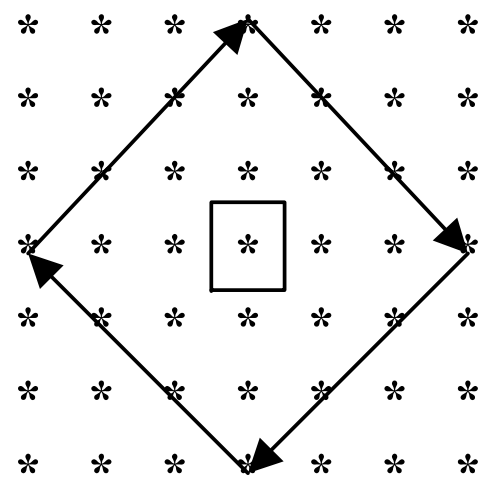
[illegible]

3.5.6 循环结构的嵌套



例题分析

把输出结果当作正方形中截取的图形，如下图：把中间的点看做是坐标系的 $(0, 0)$ 点，那么菱形就是如线所画，发现横纵坐标之和只要小于等于 M 就让它打印。假设预定义 M 值，那么所要裁剪的正方形边长为 $2*M$ ，则很快发现规律。



中间点作为 $(0,0)$ 坐标

3.5.6 循环结构的嵌套

例3.57

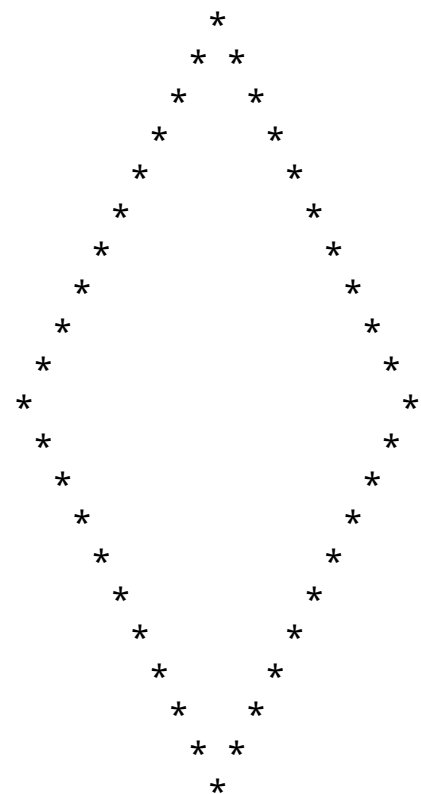
```
1 #include <stdio.h>
2 #include <math.h> //使用abs绝对值函数
3 #define M 10
4 int main()
5 {
6     int i,j;
7     for(i=-M;i<=M;i++) {
8         for(j=-M;j<=M;j++)
9             if((abs(i)+abs(j))<=M) //横纵坐标之和等于M的则全部打印
10                 printf("*");
11         else
12             printf(" ");
13         printf("\n");
14     }
15     return 0;
```

3.5.6 循环结构的嵌套



【循环程序举例】

输出图案

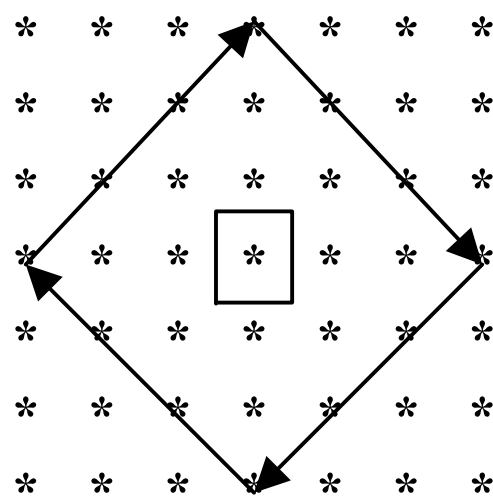


3.5.6 循环结构的嵌套



例题分析

把输出结果当作正方形中截取的图形，如下图：把中间的点看做是坐标系的 $(0, 0)$ 点，那么菱形就是如线所画，其他点依次标注，发现在线上的各点横纵坐标之和为菱形大小。假设预定义 M 值，那么所要裁剪的正方形边长为 $2 * M$ 。则很快发现规律。



中间点作为 $(0,0)$ 坐标

3.5.6 循环结构的嵌套

例3.58

```
1 #include <stdio.h>
2 #include <math.h> //使用abs绝对值函数
3 #define M 10
4 int main()
5 {
6     int i,j;
7     for(i=-M;i<=M;i++) {
8         for(j=-M;j<=M;j++)
9             if((abs(i)+abs(j))==M) //横纵坐标之和等于M的则全部打印
10                printf("*");
11         else
12             printf(" ");
13         printf("\n");
14     }
15     return 0;
```

3.5.6 循环结构的嵌套



【循环程序举例】

输出九九乘法表

```
1x1=1
2x1=2 2x2=4
3x1=3 3x2=6 3x3=9
4x1=4 4x2=8 4x3=12 4x4=16
5x1=5 5x2=10 5x3=15 5x4=20 5x5=25
6x1=6 6x2=12 6x3=18 6x4=24 6x5=30 6x6=36
7x1=7 7x2=14 7x3=21 7x4=28 7x5=35 7x6=42 7x7=49
8x1=8 8x2=16 8x3=24 8x4=32 8x5=40 8x6=48 8x7=56 8x8=64
9x1=9 9x2=18 9x3=27 9x4=36 9x5=45 9x6=54 9x7=63 9x8=72 9x9=81
```

3.5.6 循环结构的嵌套

例3.59

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j;
5     for (i=1; i<=9; i++) { //控制行
6         for (j=1; j<=i; j++) //控制每行
7             printf("%dx%d=%d ",i,j,i*j);
8         printf("\n"); //每行末尾输出1个换行
9     }
10    return 0;
11 }
```


3.5.6 循环结构的嵌套



【循环程序举例】

已知 $100 < M < 100000$ ，求 $M!$ 尾数中0的个数。

3.5.6 循环结构的嵌套



例题分析

从数学上分析， $100!$ 所包含的偶数因子比5的因子要多，而1个偶数与5相乘即产生尾数0，因此问题转化为 $M!$ 的阶乘有多少5的因子。

5、10等包含1个5的因子，而 $25=5 \times 5$ 包含2个5的因子，所以需要连续判断某数是否仍然包含5的因子。

3.5.6 循环结构的嵌套

例3.60

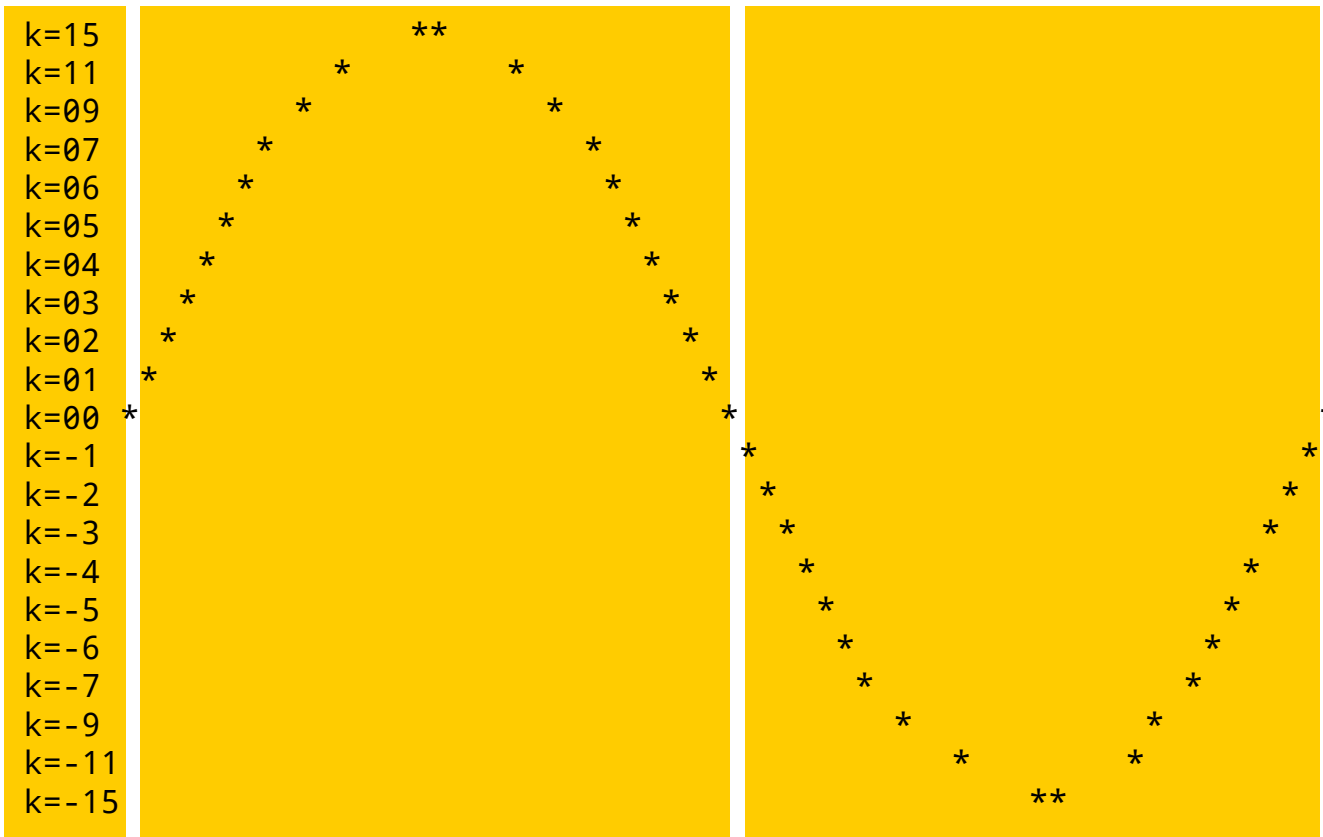
```
1  #include <stdio.h>
2  int main()
3  {
4      int i,n,m,cnt=0;
5      scanf("%d",&n);
6      for(i=5;i<=n;i=i+5) {
7          m=i;
8          while (m%5==0) { //累计包含5因子的个数
9              m=m/5;
10             cnt++;
11         }
12     }
13     printf("%d\n",cnt);
14     return 0;
15 }
```

3.5.6 循环结构的嵌套



【循环程序举例】

在屏幕上用“*”显示0-360度的正弦函数sin(x)曲线。



3.5.6 循环结构的嵌套

例3.62

```
1 #include <stdio.h>
2 #include <math.h>
3 int main()
4 {
5     int j,k ;
6     double i ;
7     for(i=1.0 ; i>=-1.0 ; i-=0.1) {
8         k = (int)(asin(i)*10);
9         printf("k=%02d ",k);
10        for(j=0 ; j<=62 ;j++)
11            if (j==k||j==62+k||j==31-k) //利用sin对称性和周期性求解
12                printf("*");
13        else
14            printf(" ");
15        printf("\n");
```

3.5.6 循环结构的嵌套

例3.62

```
16     }  
17     return 0;  
18 }
```

CP 程序设计