

第六讲

循 环

■循环

- 循环结构是在一定**条件下****反复**执行某段程序的流程结构
- 被反复执行的程序被称为**循环体**
- 每个循环都有一个**控制表达式**，循环重复一次都要对控制表达式进行**重新求值**，如果表达式**为真**（值不等于0），可以**继续**执行循环
- 3种循环语句
 - while
 - do
 - for

■ while语句

■ 格式

```
while ( expression ) statement
```

■ 示例

```
while (i < n) /* controlling expression */  
    i = i * 2; /* loop body */
```

■ 执行while语句时，**先计算**表达式的值，如果不为零，则**执行循环体**，接着**再次计算**表达式。

■ 这个过程持续进行到控制表达式的值**变为零**为止

```
while (i > 0)  
    printf("T minus %d and counting\n", i--);
```

■ 6.1 while语句

■ 例

```
i = 1;  
while (i < n) {  
    i = i * 2;  
}
```

■ 假设n=10

i = 1;

i 等于 1.

Is i < n?

是的, 继续

i = i * 2;

i 等于 2.

Is i < n?

是的, 继续

i = i * 2;

i 等于 4.

Is i < n?

是的, 继续

i = i * 2;

i 等于 8.

Is i < n?

是的, 继续

i = i * 2;

i 等于 16.

Is i < n?

不是, 退出循环

■ 6.1 while语句

控制变量初始化

```
while ( 表达式 ) {
```

语句1;

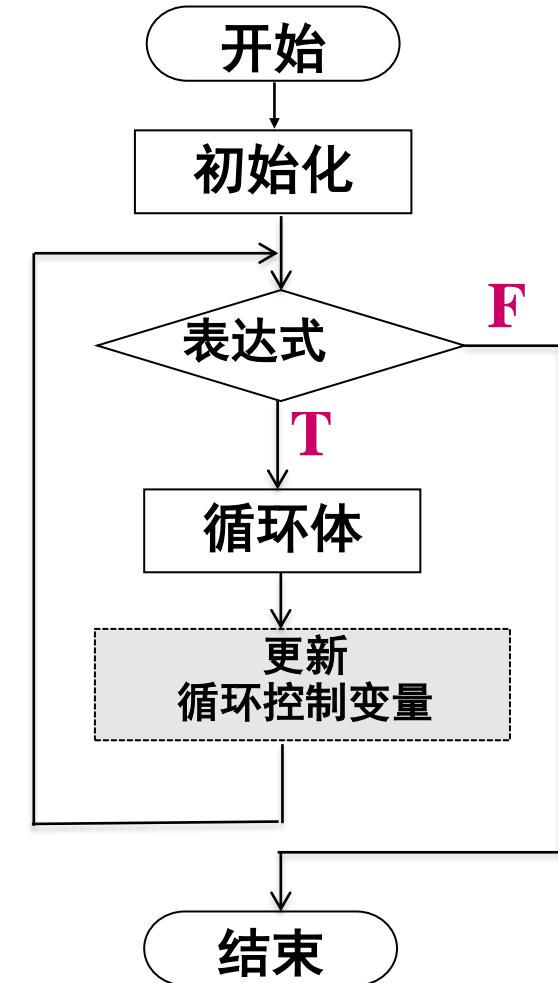
.....

语句n;

更新循环控制变量

```
}
```

```
s=0;  
i=1;  
while (i<=10){  
    s=s+i;  
    i++;  
}
```



■ 6.1 while语句

■ 示例：

- 显示平方表：用户输入一个数n，
- 然后显示出n行的输出
每行包含一个1~n的数及其平方值。

Enter number of entries in table: 5

1	1
2	4
3	9
4	16
5	25

■ 数列求和

- 对用户输入的一系列整数进行求和

```
scanf("%d", &n);
while (n != 0) {
    sum += n;
    scanf("%d", &n);
}
printf("The sum is: %d\n", sum);
```

```
/*06-01-square.c
Prints a table of squares using a while statement */
#include <stdio.h>
int main(void)
{
    int i, n;

    printf("This program prints a table of squares.\n");
    printf("Enter number of entries in table: ");
    scanf("%d", &n);

    i = 1;
    while (i <= n) {
        printf("%10d%10d\n", i, i * i);
        i++;
    }

    return 0;
}
```

■ 6.2 do语句

■ 格式

```
do statement while ( expression ) ;
```

■ 执行方式：先执行循环体，再计算控制表达式的值。如果表达式非零，再次执行循环体，然后再计算表达式值。

■ 在循环体执行后，若表达式的值为0，则终止执行do语句

```
i = 10;  
do {  
    printf("T minus %d and counting\n", i);  
    --i;  
} while (i > 0);
```

■ 6.2 do语句

■ 示例

- 计算用户输入的整数的位数
- 将输入的整数反复除以10，直到结果变为0为止；除法的次数就是所求的位数
- do更合适，因为每个整数（包括0）都至少有一位数字

```
/*06-03-numdigits.c
- Calculates the number of digits in an integer */

#include <stdio.h>
int main(void)
{
    int digits = 0, n;

    printf("Enter a nonnegative integer: ");
    scanf("%d", &n);

    do {
        n /= 10;
        digits++;
    } while (n > 0);

    printf("The number has %d digit(s).\n", digits);

    return 0;
}
```

■ 6.3 for语句

■ 格式

```
for ( expr1 ; expr2 ; expr3 ) statement
```

■ expr1, expr2, and expr3 都是表达式

■ 例

```
for ( i = 10 ; i > 0 ; i-- )  
    printf("T minus %d and counting\n", i);
```

■ for 和 while

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

表达式1 ;

while (表达式2)

语句

表达式3 ;

}

控制变量初始化

while (表达式) {

语句1;

.....

语句n;

更新循环控制变量

}

■ 6.3 for语句

■ For语句的惯用法

Counting up from 0 to n-1: `for (i = 0; i < n; i++) ...`

Counting up from 1 to n: `for (i = 1; i <= n; i++) ...`

Counting down from n-1 to 0: `for (i = n - 1; i >= 0; i--) ...`

Counting down from n to 1: `for (i = n; i > 0; i--) ...`

■ 省略表达式

```
i = 10;  
for (; i > 0; --i)  
    printf("T minus %d and counting\n", i);  
  
for (i = 10; i > 0;)  
    printf("T minus %d and counting\n", i--);
```

■ 6.3 for语句

■ C99

```
for (int i = 0; i < n; i++)
```

...

■ 逗号运算符

■ 逗号表达式: *expr1 , expr2, ...*

■ 分步执行: **计算**表达式的值1并且**扔掉**, **计算**表达式的值2并且**扔掉**, ...

```
i = 1, j = 2, k = i + j
```

as

```
((i = 1), (j = 2)), (k = (i + j))
```

■ 6.3 for语句

■ Square2.c

■ 三个表达式互相独立

```
/*06-04-square3.c
Prints a table of squares using an odd method */

#include <stdio.h>

int main(void)
{
    int i, n, odd, square;

    printf("This program prints a table of squares.\n");
    printf("Enter number of entries in table: ");
    scanf("%d", &n);

    i = 1;
    odd = 3;
    for (square = 1; i <= n; odd += 2) {
        printf("%10d%10d\n", i, square);
        ++i;
        square += odd;
    }
    return 0;
}
```

```
/*06-04-square2.c
Prints a table of squares using a for statement */
#include <stdio.h>

int main(void)
{
    int i, n;
    printf("This program prints a table of squares.\n");
    printf("Enter number of entries in table: ");
    scanf("%d", &n);

    for (i = 1; i <= n; i++)
        printf("%10d%10d\n", i, i * i);

    return 0;
}
```

■ 6.4 退出循环

■ break语句

■ 退出所在的循环语句

■ 判断一个数n是否为素数：n除以2到n-1之间的所有数，一旦有约数就**退出循环**，而不是继续执行下去

```
for (d = 2; d < n; d++)
    if (n % d == 0)
        break;
```

■ 读入用户输入，
并且在遇到特殊输入值时终止

```
for (;;) {
    printf("Enter a number (enter 0 to stop): ");
    scanf("%d", &n);
    if (n == 0)
        break;
    printf("%d cubed is %d\n", n, n * n * n);
}
```

■ 6.4 退出循环

■ continue语句

■ 退出当次循环

■ 循环读入10个非0数并求和

■ goto语句

■ 示例：账簿

■ checking.c

```
n = 0;  
sum = 0;  
while (n < 10) {  
    scanf("%d", &i);  
    if (i == 0)  
        continue;  
    sum += i;  
    n++;  
    /* continue jumps to here */  
}
```

■ 6.5 空语句

■ 除了末尾 “;” 外，**什么都没有**的语句

■ `i = 0; ; j = 1;`

■ 循环

```
for (d = 2; d < n; d++)
    if (n % d == 0)
        break;
```

```
for (d = 2; d < n && n % d != 0; d++)
    /* empty loop body */ ;
```

■ 建议：空语句**单独**一行，不要放在 `for` 那一行