

C语言程序设计

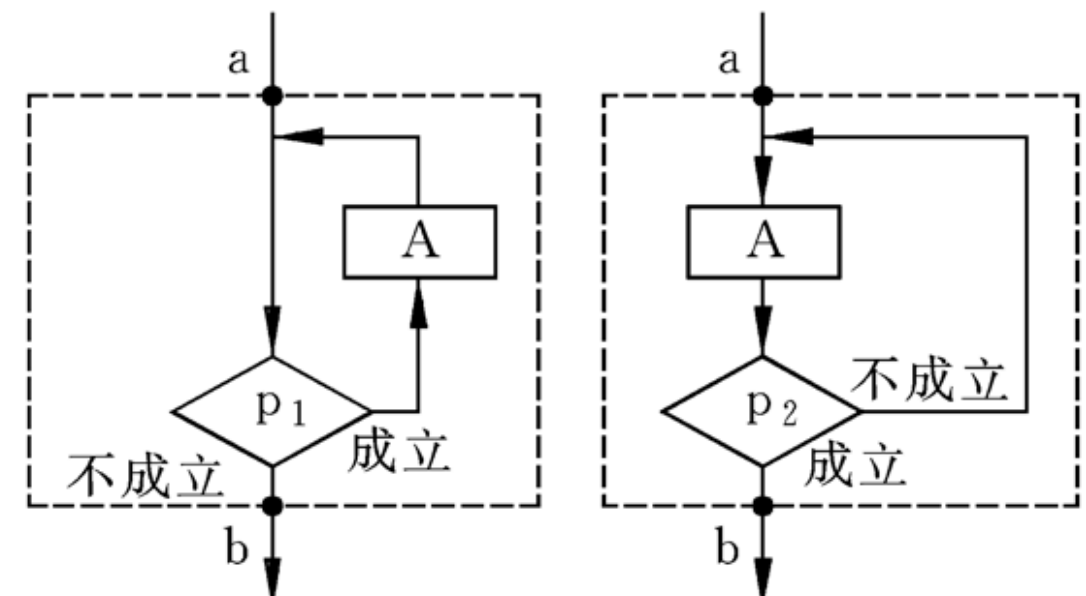
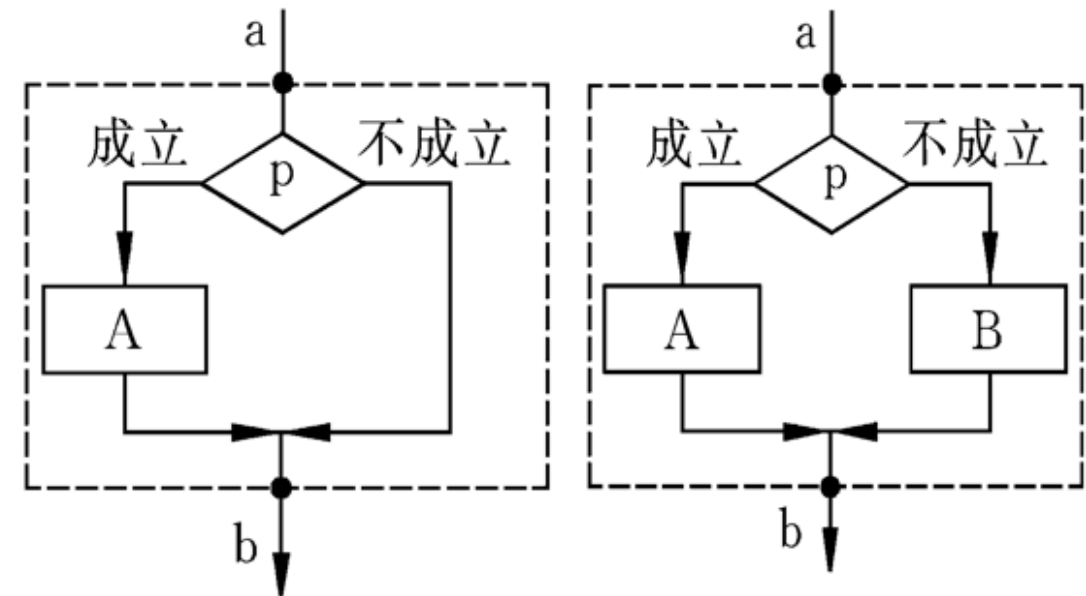
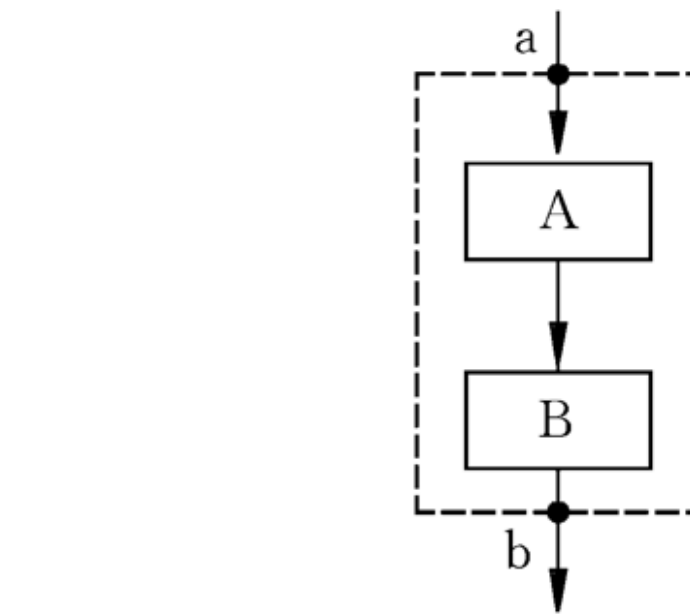
计算机科学与技术学院 王志军

循环结构程序设计



算法的基本结构

- ◎ 三种基本结构：顺序结构、选择结构、循环结构
- ◎ 循环结构：
 - ◎ 在满足某个条件时，重复执行一个/一组语句
 - ◎ 条件不满足时，结束循环



把句子抄100遍!

算法: 抄写句子

抄写一遍

抄写一遍

...

抄写一遍

玩手机去了!



伪代码：把句子抄100遍！

算法1：抄写句子

```
i <- 1    //碗里放 1 粒米
loop: 当 ( i <= 100 )时    //不到100粒
    抄写一遍
    i <- i+1 //再放 1 粒米
    转到 loop
```

玩手机去了！

算法2：抄写句子

```
i <- 100  //碗里放 100 粒米
loop: 当 ( i > 0 )时    // 碗里还有米
    抄写一遍
    i <- i-1 //拿出 1 粒米
    转到 loop
```

玩手机去了！

编程求： $s=1+2+3+ \dots +100$

◎ 步骤描述

算法1： 求和

//初始化

$s \leftarrow 0$

$s \leftarrow s+1$

$s \leftarrow s+2$

...

$s \leftarrow s+100$

输出结果 s

◎ 模式(Pattern)描述

算法2： 求和

//初始化

$s \leftarrow 0$

$i \leftarrow 1$ //碗里放 1 粒米

//循环

loop: 当 ($i \leq 100$)时 //不到100粒

$s \leftarrow s+i$

$i \leftarrow i+1$ //再放 1 粒米

转到 loop

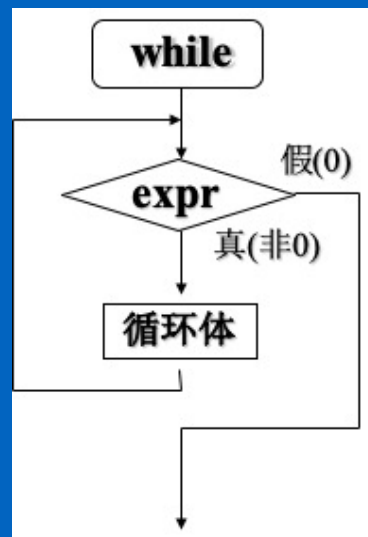
输出结果 s

C中实现循环的语句

- `while(条件){`
...
`}`
- `do {`
...
`}`
`while(条件);`
- `for(...){`
...
`}`
- `goto`

While语句

```
while(表达式){  
    循环体语句;  
}
```



- ◉ 当表达式为非0时，执行循环体
- ◉ 执行流程
 - ◉ 计算while后面的循环控制表达式的值
 - ◉ 如果表达式的值为“真”，则执行循环体，并返回步骤1
 - ◉ 如果表达式的值为“假”，则结束循环，执行循环体后面的语句

编程求： $s=1+2+3+...+100$

算法： 求和

//初始化

$s \leftarrow 0$

$i \leftarrow 1$ //碗里放 1 粒米

//循环

loop: 当 ($i \leq 100$)时 //不到100粒

$s \leftarrow s+i$

$i \leftarrow i+1$ //再放 1 粒米

 转到 loop

输出结果s

算法： 求和

//初始化

`int s=0, i=1;` //i为循环变量

//循环

`while (i<= 100) {`

`s += i;`

`i++;`

`}`

`printf("%d",s);`

分析： $s=1+2+3+ \dots +100$

```
#include <stdio.h>
int main (){
    //初始化
    //i为循环变量
    int s=0, i=1;

    //循环
    while ( i<= 100) {
        s += i;
        i++;
    }

    printf("%d",s);
}
```

循环#	i	i<=100	s
-	1	T	0
1	2	T	1
2	3	T	3
3	4	T	6
.....
99	100	T	xx
100	101	T	xx
-- 101 --	--102--	F	--xx--

编程求： $s=1+2+3+ \dots +100$

```
#include <stdio.h>
int main (){
    //初始化
    //循环变量初值
    int s=0, i=0;

    //循环次数不变
    while ( i< 100) {
        i++;
        s += i;
    }

    printf("%d",s);
}
```

循环#	i	s
-	0	0
1	1	1
2	2	3
3	3	6
.....
99	99	xx
100	100	xx
--101--	--101--	--xx--

编程求： $s=1+2+3+ \dots +100$

```
#include <stdio.h>
int main (){
    //初始化
    //循环变量初值
    int s=0, i=100;

    //循环次数不变
    while ( i > 0) {
        s += i;
        i--;
    }

    printf("%d",s);
}
```

循环#	i	s
-	100	0
1	99	100
2	98	199
3	97	297
.....
99	1	xx
100	0	xx
--101--	-- -1 --	--xx--

while语句特点

- ◎ 先思考(判断表达式),再行动(执行循环体)

- ◎ 循环体可能执行 0 次:

```
int s = 0, i = 101;
while ( i <= 100 ) {
    s = s + i;
    i++;
}
```

- ◎ 循环控制变量必须初始化

- ◎ 循环体可能执行 无限次:死循环

```
int s = 0, i = 1;
while ( i <= 100 ) {
    s = s + i;
    // 漏掉 i++;
}
```

- ◎ 循环体有若干个语句, 则用“ {} ”括起来

- ◎ 循环体可能是空语句

```
int s = 0, i = 1;
while ( i <= 100 ) ; {
    s = s + i;
    i++;
}
```

迭代法：计算100!

```
#include <stdio.h>
int main (){
    int i=1;          //循环变量赋初值
    double t=1.0;      //累乘变量赋初值
    while ( i <= 100 ){
        t=t*i;
        i++;
    }
    printf("100!=%e\n",t);
    //输出： 9.332622e+157
}
```

编写程序,输入n个学生的成绩, 求其平均值

```
#include<stdio.h>
int  main( ){
    int i,sum,n,a;
    sum = 0;
    i = 1;      //要处理的学生编号
    scanf("%d",&n); //学生个数

    while ( i <= n ){ //n为终值
        scanf("%d",&a);
        sum += a;
        i++; //下一个学生编号
    }

    printf("%d",sum/n);
}
```

显示1~10的平方

运行结果:

$$1*1=1$$

$$2*2=4$$

$$3*3=9$$

$$4*4=16$$

$$5*5=25$$

$$6*6=36$$

$$7*7=49$$

$$8*8=64$$

$$9*9=81$$

$$10*10=100$$

显示1~10的平方

运行结果:

1*1=1

2*2=4

3*3=9

4*4=16

5*5=25

6*6=36

7*7=49

8*8=64

9*9=81

10*10=100

- ◎ 自顶向下

打印 10 行信息

- ◎ 逐步求精

循环10次

每次打印 1 行信息

循环10次

打印 行号*行号

打印一个换行符

行号: 1 -> 10

打印 行号*行号

打印一个换行符

显示1~10的平方

运行结果:

1*1=1

2*2=4

3*3=9

4*4=16

5*5=25

6*6=36

7*7=49

8*8=64

9*9=81

10*10=100

行号: 1 -> 10

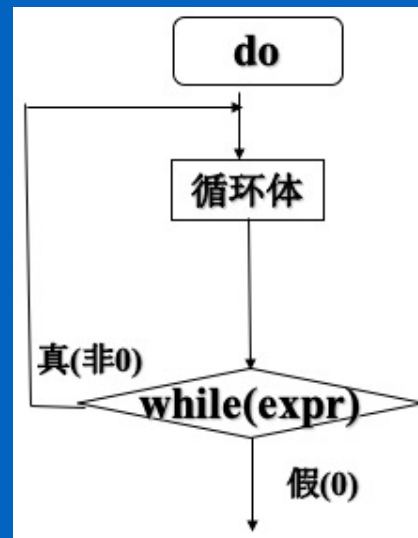
打印 行号*行号

打印一个换行符

```
#include<stdio.h>
int main( ){
    int i = 1;
    while( i <= 10) {
        printf("%d*%d=%d\n",i,i,i*i);
        i++;
    }
}
```

do-While语句

```
do{  
    循环体语句;  
}while(表达式);
```



● 执行流程

- 执行do后面的循环体语句
- 计算while后面的循环控制表达式的值
- 如果其值为“真”（非0），则返回步骤1，继续执行循环体，如果表达式的值为“假”（0），则退出此循环结构，执行循环体后面的语句。

do-while语句特点

- ◎ 先行动(执行循环体),再思考(判断表达式)
- ◎ 循环体至少执行 1 次:
- ◎ 循环体可以是一条语句,也可以是复合语句,当为一条语句时,建议也加上花括号
- ◎ while后面有一个分号(;)
- ◎ do~while可转化成while结构

```
int s = 0, i = 1;  
do{  
    s+=i;  
    i++;  
} while( i <= 100) ;
```

```
int s = 0, i = 1;  
while ( i<= 100) {  
    s = s + i;  
    i++;  
}
```

while和do~while比较

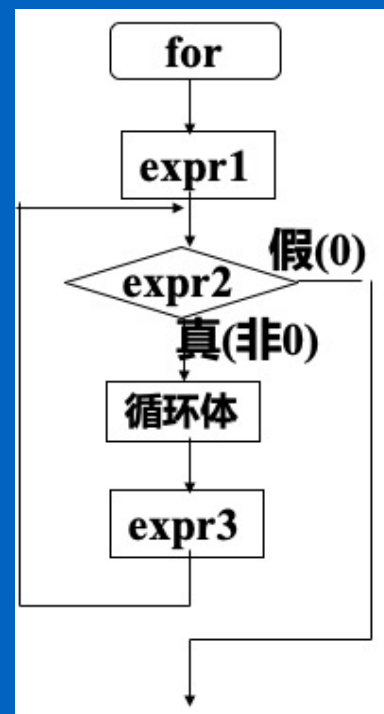
- 考虑 i=5, i=15 两种情况

```
#include<stdio.h>
int  main( ){
    int i,sum=0;
    scanf("%d",&i);
    do {
        sum+=i;
        i++;
    }while(i<=10);
    printf("%d",sum);
}
```

```
#include<stdio.h>
int  main( ){
    int i,sum=0;
    scanf("%d",&i);
    while(i<=10) {
        sum+=i;
        i++;
    }
    printf("%d",sum);
}
```

for 语句

```
for(expr1; expr2; expr3) {  
    循环体  
}
```



● 执行过程：

- (1) 计算表达式1的值
- (2) 计算表达式2的值，若其值为“真”（非0），则转（3）执行循环体；若其值为“假”（0），则转（5）结束循环
- (3) 执行循环体
- (4) 计算表达式3的值，然后转（2）判断循环条件是否成立
- (5) 结束循环，执行for循环之后的语句

● 可以转换成while结构

编程求： $s=1+2+3+...+100$

//初始化

```
int s=0, i=1; //i为循环变量
```

//循环

```
while ( i<= 100) {  
    s = s + i;  
    i++;  
}
```

```
printf("%d",s);
```

//初始化

```
int s=0, i; //i为循环变量
```

//循环

```
for (i = 1; i <= 100; i++) {  
    sum += i;  
}
```

```
printf("%d",s);
```

◉ 何时用while: 循环条件已知

```
expr1;  
while(expr2)  
{  
    循环体语句;  
    expr3;  
}
```

```
for(expr1; expr2; expr3){  
    循环体语句;  
}
```

◉ 何时用for: 循环次数已知

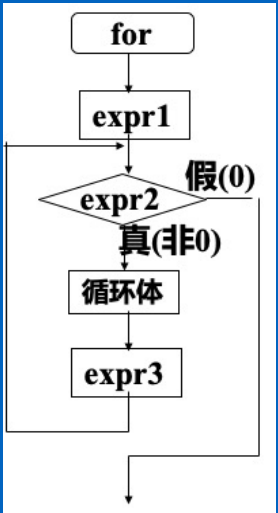
```
sum=0;  
i = 1;  
while(i <= 100){  
    sum += i;  
    i++;  
}
```

```
sum=0;  
for (i = 1; i <= 100; i++ ){  
    sum += i;  
}
```


for中 子语句执行时机与次数

```
for(expr1; expr2; expr3) {  
    循环体  
}
```

```
sum=0;  
for (i=1; i<=100; i++){  
    sum += i;  
}
```



- 若循环n次，则：
 - expr1执行 1 次；
 - expr2执行 n+1 次；
 - expr3执行 n 次；

循环#	i	s
-	1	0
1	2	1
2	3	3
3	4	6
.....
99	100	xx
100	101	xx
--101--	--102--	--xx--

◉ 用for语句输出：1 2 3 4 5

◉ expr1, expr2, expr3 类型任意，都可省略，但分号；不可省

```
int i;  
for(i = 1; i < 6; i++){ //推荐形式  
    printf("%d", i);  
}
```

```
int i;  
i = 1; //不宜实用  
for( ; i < 6; i++){  
    printf("%d", i);  
}
```

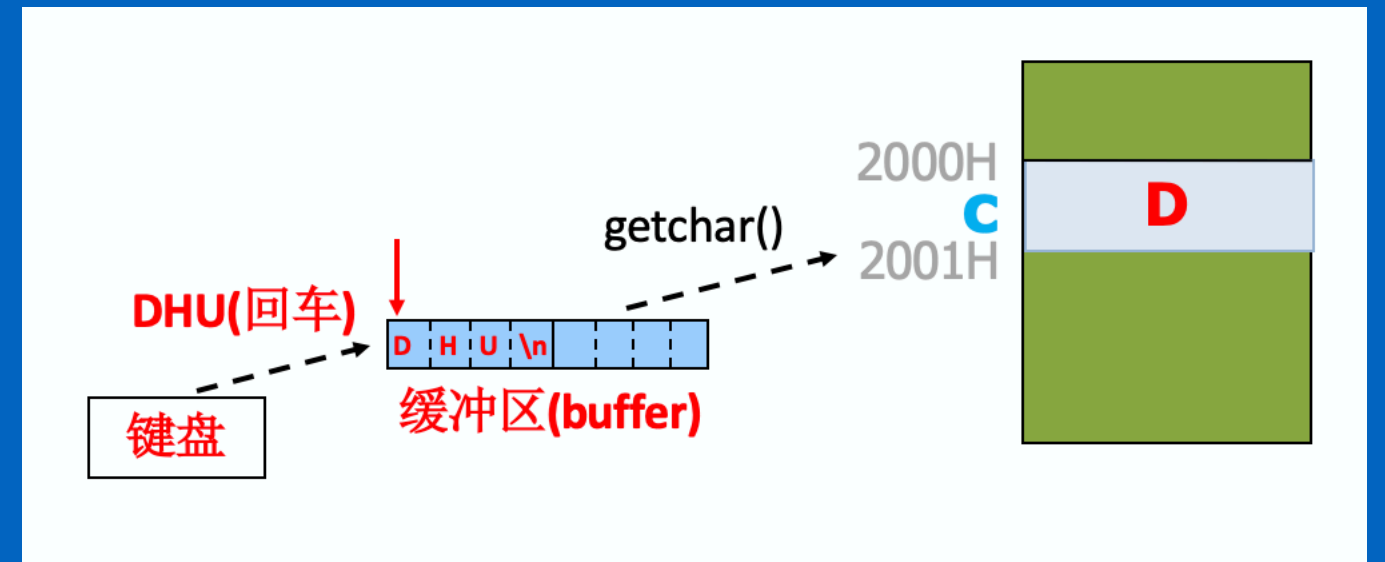
```
int i;  
i = 1; //不宜实用  
for( ; i < 6; printf("%d", i), i++){  
  
}
```

```
int i;  
for(; i < 6; ){ //不宜实用  
    printf("%d", i);  
    i++;  
}
```

◉ for(;;); 无限循环

输出用户输入的每一个字符,统计总数,直到回车结束。

```
#include<stdio.h>
int main(){
    char c;
    int i;
    for(i = 0; (c = getchar() ) != '\n' ; i++)
        putchar(c);
    printf("%d ",i);
}
```



从键盘输入一行字符，统计其中 英文字母的个数。

```
#include<stdio.h>
int main(){
    int count = 0;
    char c;
    printf("please input characters:");
    //for( ; (c = getchar()) != '\n'; ){
    while ( (c = getchar()) != '\n' ){
        if ( (c >= 'a' && c <= 'z') ||
            (c >= 'A' && c <= 'Z') )
            count++;
    }
    printf("count=%d",count);
}
```

