



第3章 顺序结构程序设计



第一节 C语言的概述

本课介绍 C 语言程序设计的基本方法和基本的程序语句。

从程序流程的角度来看，程序可以分为三种基本结构，即顺序结构、分支结构、循环结构。这三种基本结构可以组成所有的各种复杂程序。C 语言提供了多种语句来实现这些程序结构。本章介绍这些基本语句及其应用，使得对 C 程序有一个初步的认识，为后面各章的学习打下基础。



C程序是由语句构成的,可分为以下4类:

1.流程控制

1) 选择语句

if-else

(条件语句)

switch-case

(多分支选择语句)

2) 循环语句

for

(循环语句)

while

(循环语句)

do-while

(循环语句)

3) 辅助控制语句

break

(终止执行switch或循环语句)

continue

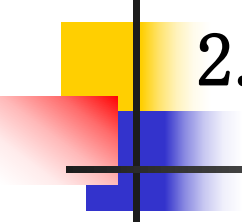
(结束本次循环)

goto

(转向语句)

return

(从函数返回语句)



2. 表达式语句

1) 赋值语句

例：

`x=8`

`/*赋值表达式*/`

`x=8;`

`/*赋值语句*/`

一个语句必须在最后出现";"

任何赋值表达式都可以加上分号成为赋值语句

例：

`a*=12 ;`

`x=y=z=8 ;`

`i++ ;`

2) 函数调用语句



例: printf("This is a example.") /*函数调用语句*/ /

scanf("%d",&c) ;

3) 空语句

例: ;

4) 复合语句

例: {

 c=a+b;

 z=3*x+y/10;

 printf("%d,%d\n", c, z) ;

 }



第二节 数据的输入输出

C语言本身不提供输入输出语句, 输入输出操作是由函数来实现的.

如: `printf()` 函数和 `scanf()` 函数



第三节 格式输入和输出

1. printf() 函数

1) 一般形式

printf("格式控制", 输出表列)

例: printf("x=%d, y=%f\n", x, y);
 格式控制 输出表列

2) 格式控制部分说明

包含3种信息:

格式说明符: %开头

转义字符: '\n' '\t' '\b'
'\101'

普通字符: printf("a=%d", a);



格式说明符

符号	作用
%d	十进制有符号整数
%u	十进制无符号整数
%f	浮点数
%s	字符串
%c	单个字符
%p	指针的值
%e	指数形式的浮点数
%x, %X	无符号以十六进制表示的整数
%o	无符号以八进制表示的整数
%g	自动选择合适的表示法

3) 输出表列部分说明

- a. `printf()` 函数中格式控制部分的“格式说明符”和“输出表列”数值类型上必须一一对应

例; `printf("a=%d, b=%ld, c=%f\n", a, b, c);`

- b. 输出表列可以是常量、变量、表达式. 如是表达式, 则 `printf()` 函数将先对其进行计算, 再输出结果

例; `main()`

```
{  
    printf("x=%8.3f", 3*4.8+9/4-sqrt(2.0));  
}
```

- c. `printf()` 函允许没有输出表列部分, 它表示输出一个字符串
格式为: `printf("输出字符串");`

例; `printf("please input a number:");`

2. scanf() 函数

1) 一般形式

scanf("格式控制",地址表列)

与printf()
函数类似

由若干个变量
地址组成(变量
前加"&")

例: **main()** /*scanf() 函数输入数据*/

```
{  
  
    int a, b;  
  
    scanf("%d, %d", &a, &b);  
  
    printf("%d, %d\n", a, b);  
  
}
```

2) 格式控制部分说明

a. 标准C语言在scanf()函数中不使用%u说明符, 对unsigned型数据以%d, %o, %x格式输入

b. 可以指定数据所占列数, 系统自动按它截取所需数据

例: `scanf("%3d%2d", &a, &b);`

运行时输入: 12345 (回车)

a得到 123 b得到 45

c. 输入数据时不能规定精度

例: `scanf("%5.2f", &x);`

3) 地址表列部分说明

a. "&"不能少

例: `scanf("%d, %d", a, b);` /*非法*/

b. 输入数据时,如果"在格式控制"字符串中,除了格式说明还有其他字符,则在输入数据时应输入对应字符

例1: `scanf("%d, %d", &a, &b);`

运行时应输入 1, 3

例2: `scanf("%d:%d", &a, &b);`

运行时应输入 1:3

例3: `scanf("x=%d y=%d", &x, &y);`

运行时应输入 x=1 y=3



c. 输入数据时, 如果"在格式控制"字符串中没有字符, 则各数据之前可以用 (空格)\Tab\Enter(回车)作为间隔符

例: `scanf ("%d%d%d%d", &a, &b, &c, &d);`

d. 输入数据为字符型(%c), 间隔符用空格\回车的话, 将不被认为是间隔符会被认为是有效数据

例: `main()
{
 char c1, c2;
 scanf ("%c%c", &c1, &c2);
 printf ("%c%c", c1, c2);
}`

运行时输入 a b /*上机*/



第四节 字符数据输入输出

1. putchar() 函数

一般形式: `putchar(参数)`

例1:

```
#include <stdio.h>

main()
{
    char c1, c2;
    c1= 'H' ;  c2= 'I' ;
    putchar(c1);
    putchar(c2);
}
```

结果 HI

2. getchar() 函数

1) 一般形式: `getchar()`

2) 使用方法:

a. 把`getchar()`函数得到的字符代码赋给一个字符型变量\整型变量

例: `int a;`

`a=getchar();`

b. 把`getchar()`函数得到的字符代码直接作为表达式的一部分, 而不赋给任何变量

例: `printf("%c", getchar());`



习题

1. 读程序

```
main()
```

```
{    int a;
```

```
    a=1234;
```

```
    printf("-%-6d\n", a);
```

```
}
```

-1234

习题

2. `main()`

`{`

`int a=-1;`

`printf("%d\t%u\t%o\n", a, a, a);`

`}`

`-1`

`65535`

`177777`

习题

3. 利用 `getchar()` 函数和 `putchar()` 函数显示由键盘输入的一个字符。

```
#include <stdio.h>
main()
{
    char c;
    c=getchar();
    putchar(c);
}
```

习题

4. 编程：使用getchar()函数，从键盘输入一个小写字母，要求输出大写字母。

```
#include <stdio.h>
main()
{
    char c1,c2;
    c1=getchar();
    c2=c1-32;
    putchar(c2);
}
```

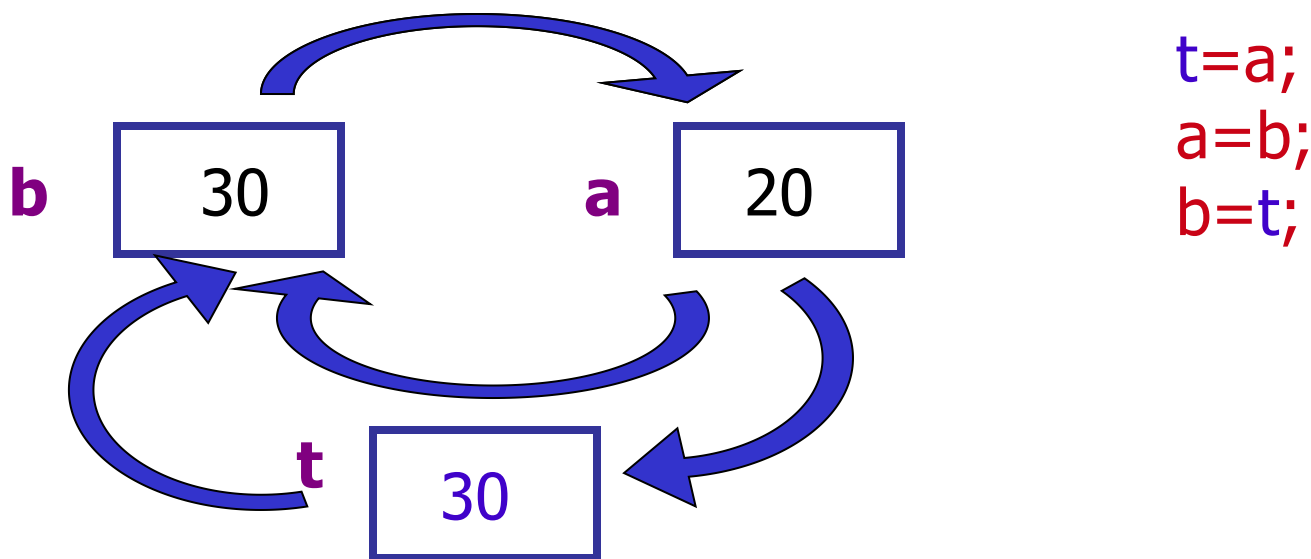
习题

5.

```
main()
{
    char s1[]="abcde";
    s1[2]='\0';
    printf("%s\n",s1);
}
```

习题

6. 从外部输入a, b两个整数，然后把c
把a和b的值进行交换，最后输出





作业

■ P56 二