

# C 程序基本结构

---

---

## □ 本章知识要点

- C 语言程序的流程控制及简单程序设计
  - 标准库函数的分类及使用
  - 常用数学函数以及标准的输入输出函数使用
  - 随机函数rand()和srand()
-

# 3.1 简单程序设计

---

## □ 3.1.1 流程控制概述

- 流程控制是指程序中语句执行的顺序，在1960年末，理论上已经证明了任何复杂的算法
  - 都可以由顺序结构、选择结构和循环结构三种基本的流程控制结构组成。对于所有的程序而言，
  - 流程控制其实都是顺序结构，也就是说程序语句总体来看就是一条接着一条按照其在程序中的
  - 位置顺序执行的。
-

- 
- C 语言提供了多种语句来实现程序结构，程序的执行部分是由语句组成，功能也是由执行语句实现。
  - C 语句可分为：表达式语句、函数调用语句、控制语句、复合语句、空语句与类。
  - 介绍这些基本语句及其在顺序结构中的应用，可以使读者对C 程序有一个初步的认识，为以后的学习打下基础
-

- 
- (1) 表达式语句：表达式语句由表达式加上分号“;”组成，运行结果可以得到表达式的值。其一般形式为：
    - 表达式;
    - 例如：`x=y+z;` 赋值语句;
    - `i++;` 自增1 语句，`i` 值增1
  - (2) 函数调用语句：由函数名、实际参数加上分号“;”组成。其一般形式为：
    - 函数名(实际参数表);
    - 函数语句的执行就可以调用函数体并把实际参数赋予函数定义中的形式参数，然后执行
    - 被调函数体中的语句（在后面函数中再详细介绍）。例如：`printf("this is a C Program");`调用库
    - 函数，输出字符串。
-

- 
- (3) 控制语句：控制语句用于控制程序的流程，C 语言由特定的语句定义符定义 9 种控制语句。可以实现程序的各种结构方式。它们可分成 3 类：
    - 条件判断语句：if 语句、switch 语句。
    - 循环执行语句：do...while 语句、while 语句、for 语句。
    - 转向语句：break 语句、goto 语句、continue 语句、return 语句。
-

---

□ (4) 复合语句：把一组语句用括号{ }括起来组成的一个语句称复合语句。

■ 在程序中把复合语句看成一个整体，相当于是单条语句，而不是多条语句。

■ 例如：

■ { x=y+z;

■ a=b+c;

■ printf("%d%d",x,a);

■ }

■ 是一条复合语句。

■ 复合语句内的各条语句都必须以分号“;”结尾，在右大括号“}”外不能加分号。

---

- 
- (5) 空语句：只由分号“;”组成的语句称为空语句。其形式为;，不产生任何操作运算，
  - 只作为形式上的语句。
  - 例如：
  - `while(getchar()!='\n')`
  - `;`
  - 本语句的功能是，只要从键盘输入的字符不是回车则重新输
-



---

## □ 3.1.2 简单程序设计举例

- 一个简单的程序仅包含一个main()函数并且程序流程是顺序结构，程序的执行部分通常包含3种基本的语句：输入语句、输出语句、赋值语句，或用于计算的其他表达式语句。
-

- 例3-1 输入三角形的三边长，求三角形面积。
- 分析：已知三角形的三边长a， b， c， 则该三角形的面积公式为：
$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

□ 其中 $s = (a+b+c)/2$ ， 所以只需要输入三个边， 就可以得到三角形的面积。

```
■ #include<math.h>
■ #include<stdio.h>
■ int main()
■ {
■ float a,b,c,s,area;
■ scanf("%f,%f,%f",&a,&b,&c); /* 输入边长a, b, c */
■ s=1.0/2*(a+b+c);
■ area=sqrt(s*(s-a)*(s-b)*(s-c)); /* 计算三角形面积 */
■ printf("a=%7.2f,b=%7.2f,c=%7.2f,s=%7.2f\n",a,b,c,s);
■ printf("area=%7.2f\n",area); /* 输出三角形面积 */
■ return 0;
■ }
```

- 
- 程序的运行情况为:
  - 请输入三角形边（如3,5,6）:3,4,5
  - $a=□□□3.00, b=□□□4.00, c=□□□5.00, s$   
 $=□□□6.00$
  - $area=□□□6.00$
-

- 
- 例3-2 将字符'C'、'h'、'i'组成的字符串Chi加密（将每位字符ASCII值加6后变成其他字符的ASCII值）后输出并显示。
  - 分析：利用字母的ASCII码值可以进行加减运算特性，在ASCII码运算后再进行输出即可将字母进行加密转换。本例题是将每位字符ASCII值加6后变成其他字符的ASCII值，然后利用printf()进行输出。
-

```
# include <stdio.h>
int main()
{
/* 定义字符型变量，并给它们赋初值 */
char c1, c2, c3, c4, c5, c6, c7;
c1 = 'C';
c2 = 'h';
c3 = 'i';
/* 输出原码 */
printf("原码是: %c%c%c \n", c1, c2, c3);
/* 对字符进行译码运算 */
c1 = c1 + 6;
c2 = c2 + 6;
c3 = c3 + 6;
/* 输出译码结果 */
printf("密码是: %c%c%c \n", c1, c2, c3);
return 0;
}
```

- 
- 运行情况如下
  - 原码是：Chi
  - 密码是：Ino
-

## 3.2 C 语言标准库函数

---

- ❑ C 源程序是由函数组成的。虽然在前面程序中都只有一个主函数`main()`，但实际使用程序往往由多个函数组成。函数是C源程序的基本模块，相当于其他高级语言的子程序。
  - ❑ C 程序的全部工作都是由各式各样的函数完成的，所以通常也会把C语言称为函数式语言。
  - ❑ C 语言提供了极为丰富的库函数（如Turbo C，Microsoft C 都提供了三百多个库函数），库函数由C编译环境提供，用户无须定义，只需在程序的首部使用预处理命令`#include` 包含有该函数原型的头文件，即可在程序中直接调用，
-

---

## □ 3.2.1 C 标准库函数的分类

□ C 语言丰富的库函数，从功能角度分为以下7 类。

### □ (1) 数学函数。

- 此类函数主要功能是由于数学函数计算。数学函数原型在头文件`math.h`，使用此类函数前，必须在程序首部将`math.h` 头文件包含进来方可使用。本章3.2.2 节常用数学库函数将详细讲解该类函数，在此不再赘述。

### □ (2) 字符判断和转换函数。

- 此类函数用于对字符按ASCII 码分类为：字母、数字、控制字符、分隔符、大小写字母等，也可以使用字符转换函数将参数转换为需要的大小写格式。所有的字符函数都包含在`ctype.h` 头文件中，使用字符函数前都必须在程序的首部将头文件`ctype.h` 包含到程序中。
-



---

## □ (3) 输入输出函数。

- 此类函数主要功能是用于完成数据输入输出功能。函数原型都在头文件`stdio.h`中,使用此类函数前,必须在程序的首部将头文件`stdio.h`包含到程序中。本章3.2.3节、3.2.4节将详细讲解该类函数,在此不再赘述。

## □ (4) 字符串函数。

- 此类函数主要功能是用于字符串操作和处理。字符串函数原型都在头文件`string.h`中,使用此类函数前,必须在程序的首部将头文件`string.h`包含到程序中进来方可使用。
-

---

## □ (5) 动态存储分配（内存管理）函数。

- 此类函数主要功能是由于内存管理。内存管理函数原型在头文件`alloc.h`，使用此类函数前，必须在程序首部将`alloc.h`头文件包含进来方可使用。后面第9章指针的高级章节将详细讲解该类函数，在此不再赘述。

## □ (6) 日期和时间函数。

- 此类函数主要功能是获得系统时间或对得到的时间进行格式化等操作。日期和时间函数原型在头文件`time.h`，使用此类函数前，必须在程序首部将`time.h`头文件包含进来方可使用。
-

---

## □ (7) 其他函数。

■ 此类函数主要功能是由于其他各种功能。其他函数原型在头文件 `stdlib.h`，使用此类函数前，首先必须在程序头部使用预处理命令 `#include <stdlib.h>` 将头文件包含进来方可使用。可以利用此类函数中的 `rand()` 函数和 `srand()` 函数来取得随机数，本章 3.2.6 节具体讲解随机函数使用方法。

□ 注意：在 C 语言程序中用预处理命令将头文件包含到程序的形式有两种，例如，针对头文件 `stdio.h` 来说预处理命令的形式可以为 `#include <stdio.h>` 或 `#include "stdio.h"`，使用尖括号表示编译时会先在系统的 `include` 目录里查找，若找不到才会到源代码所在目录进行搜索；使用双引号则相反。建议对系统提供的头文件使用尖括号，自己编写的头文件使用双引号。

---

---

## □ 3.2.2 常用数学库函数

- C 语言提供的数学库函数可以解决一些只用算术运算符不能完成的问题。数学函数原型都包含在 `math.h` 头文件中。除了简单的数学函数，程序开发常用的三角函数和对数函数如表 3-1 所示。

**表3-1**常用三角函数、对数数学库函数

函数名	函数和形参类型	功能	返回值	说明
sin	double sin( x) double x;	计算 <b>sin x</b> 的值	计算结果	x的单位为弧度
cos	double cos(x) double x;	计算 <b>cos(x)</b> 的值	计算结果	x的单位为弧度
exp	double exp(x) double x;	求 <b>e<sup>x</sup></b> 的值	计算结果	
log	double log(x) double x;	求 <b>log<sub>e</sub>x</b> ,即 <b>lnx</b>	计算结果	<b>x&gt;0</b>
log10	double log10(x) double x;	求 <b>log<sub>10</sub>x</b>	计算结果	<b>x&gt;0</b>
pow	double pow(x,y) double x,double y	计算 <b>x<sup>y</sup></b> 的值	计算结果	
pow10	double pow10 (x) int x	计算 <b>10<sup>x</sup></b> 的值	计算结果	

## 例3-5 打印出三角函数和对数函数的结果。

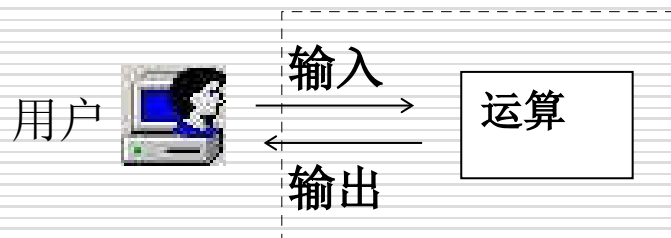
```
 #include<math.h>
 #include<stdio.h>
 int main()
 {
 printf("三角函数: \n");
 printf("三角函数 cosine of 1 is %.3f\n",cos(1));
 printf("三角函数 sine of 1 is %.3f\n",sin(1));
 printf("三角函数 tangent of 1 is %.3f\n",tan(1));
 printf("\n");
 printf("\n");
 printf("对数函数: \n");
 printf("对数函数e 的1 次方is %.3f\n",exp(1));
 printf("2 的自然对数函数is %.3f\n",log(2));
 printf("2 的以10 为底的对数函数is %.3f\n",log10(2));
 return 0;
 }
```

### 例3-6 用pow()函数和pow10()函数，分别求10 的3 次幂的值。

- ❑ #include<math.h>
  - ❑ #include<stdio.h>
  - ❑ int main()
  - ❑ {
  - ❑ double p=3.0;
  - ❑ printf("pow()函数10^%lf is %lf\n",p,pow(10,p)); /\* pow()函数输出10 的3 次方的值\*/
  - ❑ printf("pow10 函数10^%lf is %lf\n",p,pow10(p)); /\* pow10()函数输出10 的3 次方的值\*/
  - ❑ return 0;
  - ❑ }
  
  - ❑ 运行结果:
  - ❑ pow()函数10^3.000000 is 1000.000000
  - ❑ pow10()函数10^3.000000 is 1000.000000
-

### □ 3.2.3 标准输出函数

- 一般C语言程序可分为：数据输入、计算处理和输出处理结果3部分，其中数据的输入与输出是程序的重要组成部分，是程序与用户之间交互的界面如图3-1所示





- 
- 其他的高级语言都提供了输入与输出语句，而C语言本身没有输入和输出语句，为实现数据的输入和输出，其库函数提供了一组输入和输出函数，
  - 函数scanf()和printf()来完成数据的格式输入和输出操作，其作用是向默认的输入设备（键盘）和输出设备（终端或显示器）输入或输出若干个任意类型的数据。
-

---

## □ 1. printf()函数的功能

- printf()函数称为格式输出函数，最末一个字母f即为“格式”（format）之意，格式输出函数printf()的一般调用形式为：
  - **printf("格式控制字符串",输出项表列);**
  - 如:printf("a=%d, b=%d\n",a,b);
-

# `printf("格式控制字符串",输出项表列);`

- ❑ `printf()` 函数功能是按用户所指定“格式控制字符串”的格式，将指定的输出项表列数据输出到标准输出设备（通常为显示器）。
- ❑ 格式控制字符串是使用一对双引号括起来的字符串，格式字符串用于指定后面各个输出项的输出格式。
- ❑ 输出项表列用于指定输出内容，它通常由一个或多个输出项构成，当有多个输出项时，输出项之间应使用逗号“,”分隔，输出项可以是常数、变量或表达式。

---

## □ 2. 输出格式

- 输出格式由格式控制字符串加以规定，将输出项表列相对应的输出项以指定的格式进行输出。
  - 格式控制字符串有“格式字符”和“普通字符（包括转义字符序列）”两种字符组成，普通字符串在输出时原样输出（或执行），普通字符主要是在显示中起提示作用。
  - 格式字符形式为：`%[附加格式说明符]格式字符`
  - 例如，`%d`，`%10.2f` 等。
-

## (1) 格式字符。

- 最简单的格式说明符是以%开头后面跟上一个特定的字母，用来与输出项的数据类型相匹配
- 如：
  - “%d”表示按十进制整型输出。
  - “%ld”表示按十进制长整型输出。
  - “%c”表示按字符型输出一个字符。
  - “%s”表示按实际宽度输出一个字符串

## 表3-2格式说明符

格式字符	功能
d	输出带符号10进制形式整数(正数不输出符号)
o	输出无符号8进制形式整数(不输出前缀0)
x,X	输出无符号16进制形式整数(不输出前缀0x)
u	输出无符号10进制形式整数
f	输出单、双精度小数形式实数(6位小数)
c	输出单个字符
s	输出一串字符串
e,E	以指数形式输出单、双精度实数(尾数含1位整数, 6位小数, 指数至多3位)
g,G	以%f或%e中输出宽度较小的格式输出单、双精度实数, 不输出无意义的0

---

## □ 1) 整型数据格式输出。

- `%d`: 输出十进制基本整型数据。
  - 例如, 设整型数据 `a=12;b=13;printf("%d,%d",a,b);`
  - `%ld`: 输出十进制长整型数据。
  - 例如, `long int a=1234567;printf("%ld",a);`
  - `%u` (或 `%lu`): 输出无符号十进制基本整型数据 (或长整型数据)。
  - `%o` (或 `%lo`): 以无符号八进制格式输出整型数据 (或长整型数据)。
  - `%x` (或 `%lx`): 以无符号十六进制格式输出整型数据 (或长整型数据)。
-

---

## □ ②实型格式格式输出

- C语言程序的浮点数有10进制小数和科学计数法两种输出形式，对于单精度和双精度浮点数来说，它们的格式说明符完全相同。
    - %f: 以10进制小数形式输出单精度、双精度浮点数。
    - %e: 以科学计数法形式输出单精度、双精度浮点数。
    - %g: 根据浮点数的大小，自动选用%f或%e格式中输出宽度较短的一种格式，且不输出无意义的零。
-



- 
- **【例3.7】** 以10进制小数形式输出单精度、双精度浮点数
  - `#include <stdio.h>`
  - `int main()`
  - `{`
  - `double x=3.1415;`
  - `float y=15.725;`
  - `printf("x=%f,y=%f\n",x,y);`
  - `printf("x=%e,y=%e\n",x,y);`
  - `printf("x=%f, x=%e, x=%g" x, x, x );`
  - `return 0;`
  - `}`
  - 程序运行结果为
  - `x=3.141500,y=15.725000`
  - `x=3.141500e+000,y=1.572500e+001`
  - `x=3.141500, x=3.141500e+000, x=3.1415`
-

---

### □ ③字符型格式说明符

- %c: 输出一个字符。
- char c;
- c='s';
- printf("%c, ",c);
- 则程序输出结果为: s

### □ ④字符串格式说明符

- %s: 按实际宽度输出一个字符串。  
printf("%s","hello);
  - 输出结果为: hello
-

---

## □ (2) 转义字符

- 转义字符作为格式控制字符串中的非格式字符，由“\”和一个特定的字母组成，用于输出某些特殊字符和不可见字符。常用的转义字符符如表3-3所示。

### 表3-3转义字符

转义字符形式	功能
\n	换行
\t	横向跳格（即跳到下一个输出区-占8列）
\v	竖向跳格
\b	退格(不换行)
\r	回车
\f	走纸换页
\\	反斜杠字符“\”
\'	单引号(撇号)字符
\ddd	1到3位8进制数所代表的字符
\xdd	1到2位16进制数所代表的字符

---

### □ (3) 附加格式说明符

- 在%和格式符之间的附加格式说明符，用于指定输出时的对齐方向、输出数据的宽度、小数部分的位数等要求，附加格式说明符可以是其中之一或多个字符的组合。常用的附加说明符如表3-4所示。

## 表3-4附加格式说明符

附加说明符	意义
m(m为正整数)	为域宽描述符, 数据输出宽度为m, 若实际位数多于定义的宽度, 则按实际位数输出, 若实际位数少于定义的宽度则补以空格或0
.n(n为正整数)	为精度描述符, 对实数, n为输出的小数位数, 若实际位数大于所定义的精度数, 则截去超过的部分;对于字符串, 表示输出n前各字符。
l	表示整型按长整型量输出如%ld,%lx,%lo, 对实型按双精度型量输出如%lf,%le。
h	表示按短整型量输出, 如%hd,%hx, %hdo,%hu
-	数据左对齐输出, 右边填充空格, 无-时默认右对齐输出
+	输出符号(正号或负号)
0	表示数据不足最小输出宽度时, 左补零;
空格	输出值为正时冠以空格, 为负时冠以负号
#	对c,s,d,u类无影响; 对o类,在输出时加前缀0; 对x类,在输出时加前缀0x; 对e,g,f类当结果有小数时才给出小数点

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/926102233120010235>