



计算机导论与程序设计——第6篇

数据对象与计算

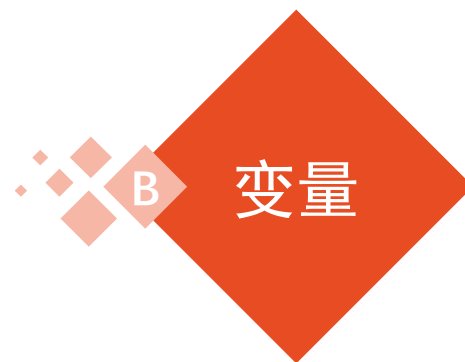
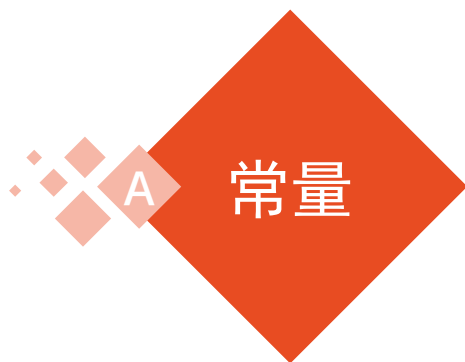
Computer Introduction and Programming

学习目标



- 熟悉C语言字符集、保留字集、标识符、算符等基本词法元素
- 熟悉C语言数据类型，掌握变量声明方法
- 掌握C语言运算符、表达式与计算过程
- 熟悉语句与复合结构
- 掌握键盘输入和屏幕输出等交互方法
- 熟悉数值计算类C程序的一般形式

在计算机高级语言中，数据的两种表现形式：



常量

在程序运行过程中，其值不能被改变的量

- 1 整型常量 1000,12345,0,-345
- 2 实型常量 **小数形式**123.4; **指数形式**1.234e2
- 3 字符常量 **普通字符**'a','Z','#'; **转义字符**'\n'
- 4 字符串常量 "123","boy"
- 5 符号常量 #define PI 3.1416 //注意行末没有分号

数据的表现形式及其运算

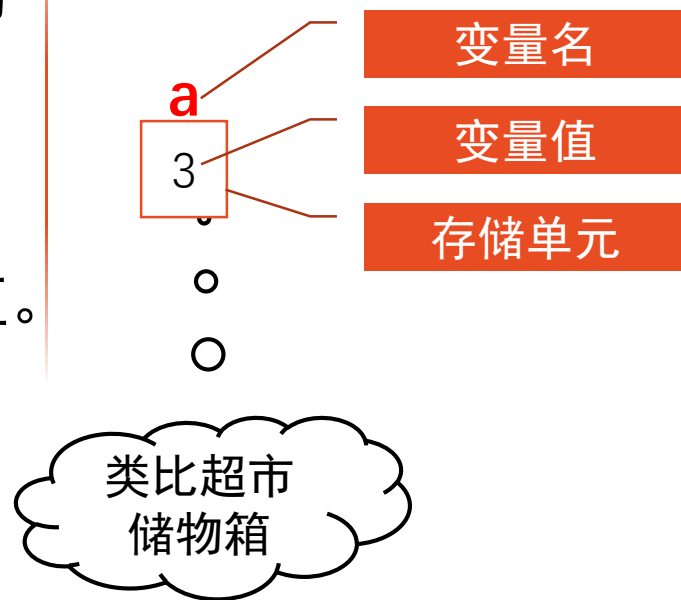
变量

在程序运行期间，变量的值是可以改变的。

变量代表一个有名字的、具有特定属性的一个存储单元。

变量用来存放数据，也就是存放变量的值。

变量必须先定义，后使用。



数据的表现形式及其运算

标识符

标识符就是一个对象的名字。用于标识变量、符号常量、函数、数组、类型等

标识符只能由字母、数字和下划线3种字符组成，且第1个字符必须为字母或下划线

注意

- 变量名中区分大小写字母
- 不能使用关键字作为变量名
- 变量的名字应该尽量反映变量在程序中的作用与含义

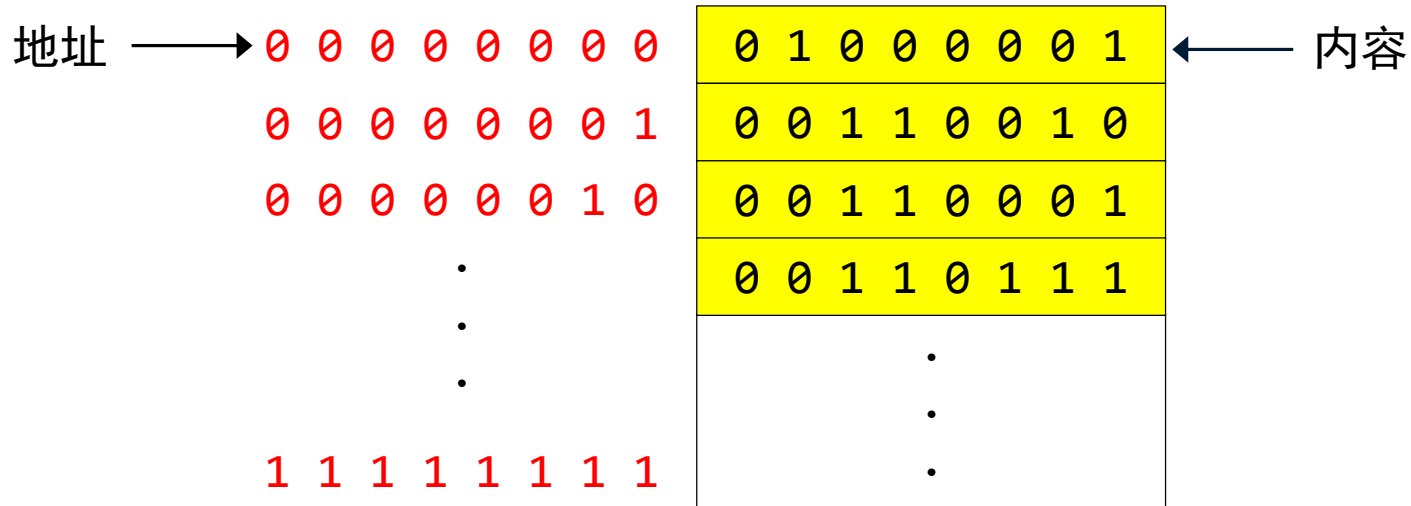
见名知义

C语言中的关键字

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

数据类型。

为什么需要定义数据类型？

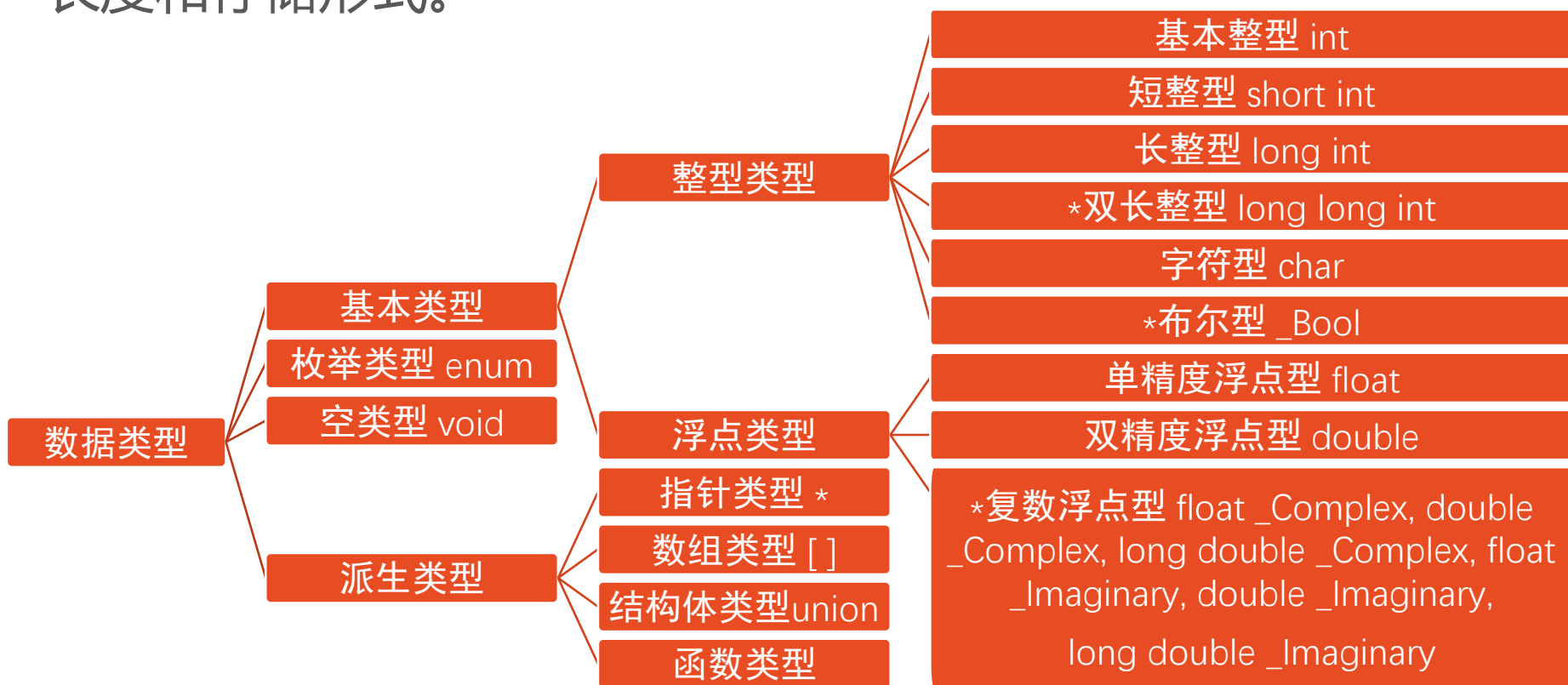


- 存储信息的基本单位是位(bit)
- 每8位二进制数合在一起称为一个字节(Byte)
- 存储器的一个存储单元一般存放一个字节的的信息
- 每个存储单元都有唯一的编号，称为地址

高位 低位	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	EXT	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	—	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

数据类型

所谓类型，就是对数据分配存储单元的安排，包括存储单元的长度(占多少字节)以及数据的存储形式。不同的类型分配不同的长度和存储形式。



*注：C99新引入

计算机中带符号整型数的表示：补码

正整数的补码就是此数的二进制形式，5的补码：

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

-5的补码：

①原码

1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

②反码

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

③补码

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

整型数据

整型数据类型	缺省形式的整型数据类型	字节数	取值范围
[signed] int	<u>int</u>	4	$-2^{31} \sim 2^{31}-1$
unsigned [int]	unsigned <u>int</u>	4	$0 \sim 2^{32}-1$
[signed] short [int]	short	2	$-2^{15} \sim 2^{15}-1$
unsigned short [int]	unsigned short	2	$0 \sim 2^{16}-1$
[signed] long [int]	long	4	$-2^{31} \sim 2^{31}-1$
unsigned long [int]	unsigned long	4	$0 \sim 2^{32}-1$
[signed] long long [int]	long <u>long</u>	8	$-2^{63} \sim 2^{63}-1$
unsigned long long [int]	unsigned long <u>long</u>	8	$0 \sim 2^{64}-1$

说明：C标准没有具体规定各种类型数据所占用存储单元的长度，只要求 $\text{sizeof}(\text{short}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long}) \leq \text{sizeof}(\text{long long})$ ，具体由各编译系统自行决定的。
 sizeof 是测量类型或变量长度的运算符。

字符型数据

ASCII字符集包括:

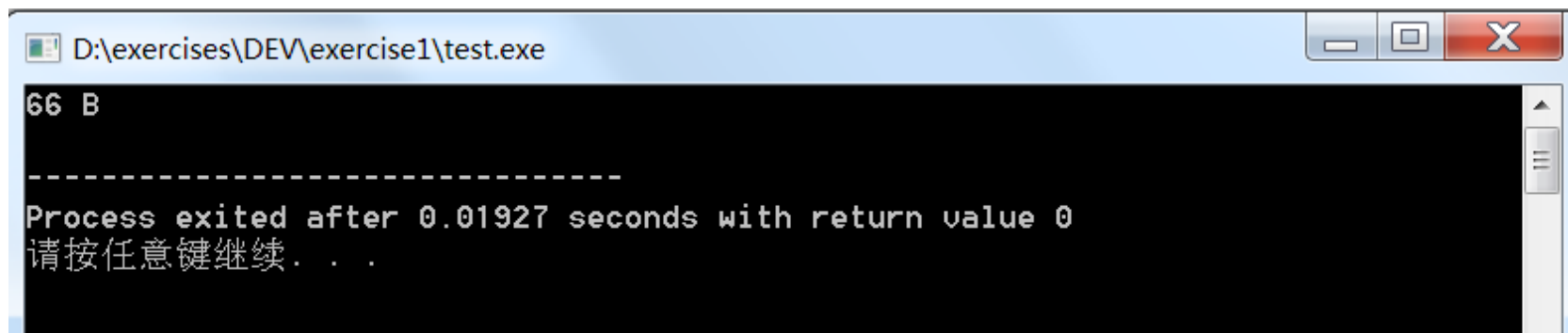
- 字母: 大写英文字母A~Z, 小写英文字母a~z
- 数字: 0~9
- 专门符号: 29个,包括
- ! " # & ' () * + , - . / : ; < = > ? [\] ^ _ ` { | } ~
- 空格符: 空格、水平制表符(tab)、垂直制表符、换行、换页(form feed)
- 不能显示的字符: 空(null)字符(以'\0'表示)、警告(以'\a'表示)、退格(以'\b'表示)、回车(以'\r'表示)等

高位 低位	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	EXT	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	—	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

字符变量是用类型符char定义字符变量。

```
char c = 'B'; //定义c为字符型变量并使初值为字符'B'
```

```
printf("%d %c\n",c,c); //用 “%d”格式输出十进制整数66，用  
“%c”格式输出字符'?'
```

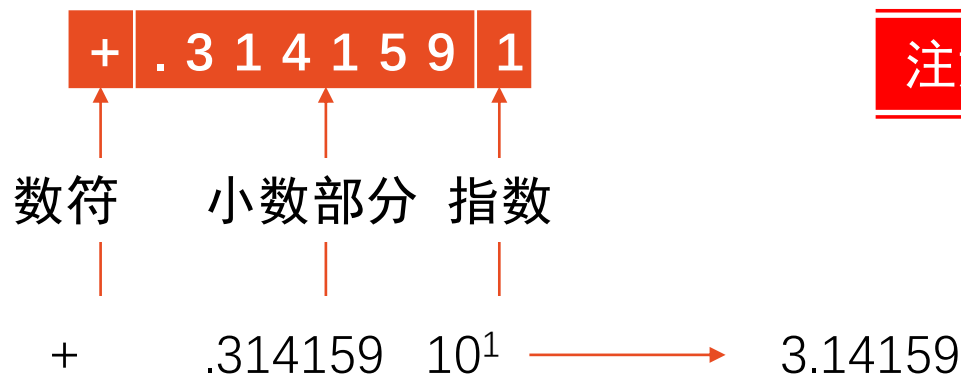


```
D:\exercises\DEV\exercise1\test.exe
66 B
-----
Process exited after 0.01927 seconds with return value 0
请按任意键继续...
```

浮点型数据

$$3.14159 = 3.14159 * 10^0 = 0.314159 * 10^1 = 314.159 * 10^{-2}$$

由于小数点位置可以浮动，所以实数的指数形式称为**浮点数**。
浮点数类型包括float(单精度浮点型)、double(双精度浮点型)、long double(长双精度浮点型)。



注意

小数部分占的位（bit）数愈多，数的有效数字愈多，精度也就愈高。指数部分占的位数愈多，则能表示的数值范围愈大。

实型数据

类型	字节数	有效数字	数值范围（绝对值）
float	4	6	0以及 $1.2 \times 10^{-38} \sim 3.4 \times 10^{38}$
double	8	15	0以及 $2.3 \times 10^{-308} \sim 1.7 \times 10^{308}$
long double	8	15	0以及 $2.3 \times 10^{-308} \sim 1.7 \times 10^{308}$
	16	19	0以及 $3.4 \times 10^{-4932} \sim 1.1 \times 10^{4932}$

运算符和表达式

C语言的运算符

算术运算符	+ - * / %	逗号运算符	,
关系运算符	> < == >= <= !=	指针运算符	* &
逻辑运算符	! &&	求字节数运算符	sizeof
位运算符	<< >> ~ ^ &	强制类型转换符	(type)
赋值运算符	=	分量运算符	· ->
条件运算符	? :	下标运算符	[]

运算符的三要素

功能、优先级、结合方向。



算术表达式和运算符的优先级与结合性

用算术运算符和括号将运算对象（也称操作数）连接起来的、符合 C 语法规则的式子称为**C算术表达式**。

运算对象包括常量、变量、函数等。

C语言规定了运算符的**优先级**(例如先乘除后加减)，还规定了运算符的**结合性**。

在表达式求值时，先按运算符的优先级别顺序执行。



算术表达式和运算符的优先级与结合性

当在一个运算对象两侧的运算符的优先级别相同时，则按规定的“结合方向”处理。C语言规定了各种运算符的结合方向（结合性），“自左至右的结合方向”又称“左结合性”，即运算对象先与左面的运算符结合。相反“自右至左的结合方向”称为“右结合性”。

C语言的运算符-算术运算符

运算符	功能	运算对象个数	优先级	结合方向
* / %	乘	2	3	自左至右
	除	2	3	
	取模 (求余)	2	3	
+ -	加	2	4	自左至右
	减	2	4	
=	赋值	2	14	自右至左
++ -- -	自增, 加1	1	2	自右至左
	自减, 减1	1	2	
	负号	1	2	



两个实数相除的结果是双精度实数，两个整数相除的结果为整数



%运算符要求参加运算的运算对象(即操作数)为整数，结果也是整数

自增 (++) 自减 (--) 运算符

`++i, --i`

在使用 `i` 之前, 先使 `i` 的值加/减1

`i++, i--`

在使用 `i` 之后, 使 `i` 的值加/减1

前置加(减)和后置加(减): 都是使该变量的值增加1或减少1。
当它们与其他运算结合在一个表达式中时, 其运算值就不同了。

建议谨慎使用 `++` 和 `--` 运算符, 只用最简单的形式, 即 `i++`, `i--`, 且把它们作为单独的表达式。

C语言的运算符-关系和逻辑运算符

运算符	功能	运算对象个数	优先级	结合方向
>	大于	2	6	自左至右
>=	大于等于	2	6	
<	小于	2	6	
<=	小于等于	2	6	
==	等于	2	7	自左至右
!=	不等于	2	7	
!	逻辑非（取反）	1	14	自右至左
&&	逻辑与	2	11	自左至右
	逻辑或	2	12	

C语言的运算符-逻辑运算真值表

p	q	p&&q	p q	!p
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

C语言的运算符-其它运算符

运算符	功能	运算对象个数	优先级	结合方向
<div>() [] -> .</div>	<div>圆括号 下标运算 成员运算 成员运算</div>		<div>1 1 1 1</div>	自左至右
<div>(类型) * & sizeof</div>	<div>类型转换 指针运算 地址运算 求类型长度</div>	<div>1 1 1 1</div>	<div>2 2 2 2</div>	自右至左

【例6.1】给定一个大写字母，要求用小写字母输出。

高位 低位	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	EXT	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	—	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

【例6.1】给定一个大写字母，要求用小写字母输出。

解题思路：字符数据以ASCII码存储在内存中，形式与整数的存储形式相同。所以字符型数据和其他算术型数据之间可以互相赋值和运算。大小写字母之间的关系是：同一个字母，用小写表示的字符的ASCII代码比用大写表示的字符的ASCII代码大32。

```
1 #include <stdio.h>
2 int main()
3 {   char c1,c2;
4     c1='A'; //将字符'A'的ASCII代码放到c1变量中
5     c2=c1+32; //得到字符'a'的ASCII代码，放在c2变量中
6     printf("%c\n",c2); //输出c2的值，是一个字符
7     printf("%d\n",c2); //输出c2的值，是字符'a'的ASCII代码
8     return 0;
9 }
```

在运算时不必用户干预，系统自动进行的类型转换。

自动类型转换

类型转换

强制类型转换

当自动类型转换不能实现目的时，可以用强制类型转换。

强制类型转换运算符

(类型名)(表达式)

(double)a 将 a 转换成double型

(int)(x+y) 将x+y的值转换成int型

(float)(5%3) 将5%3的值转换成float型

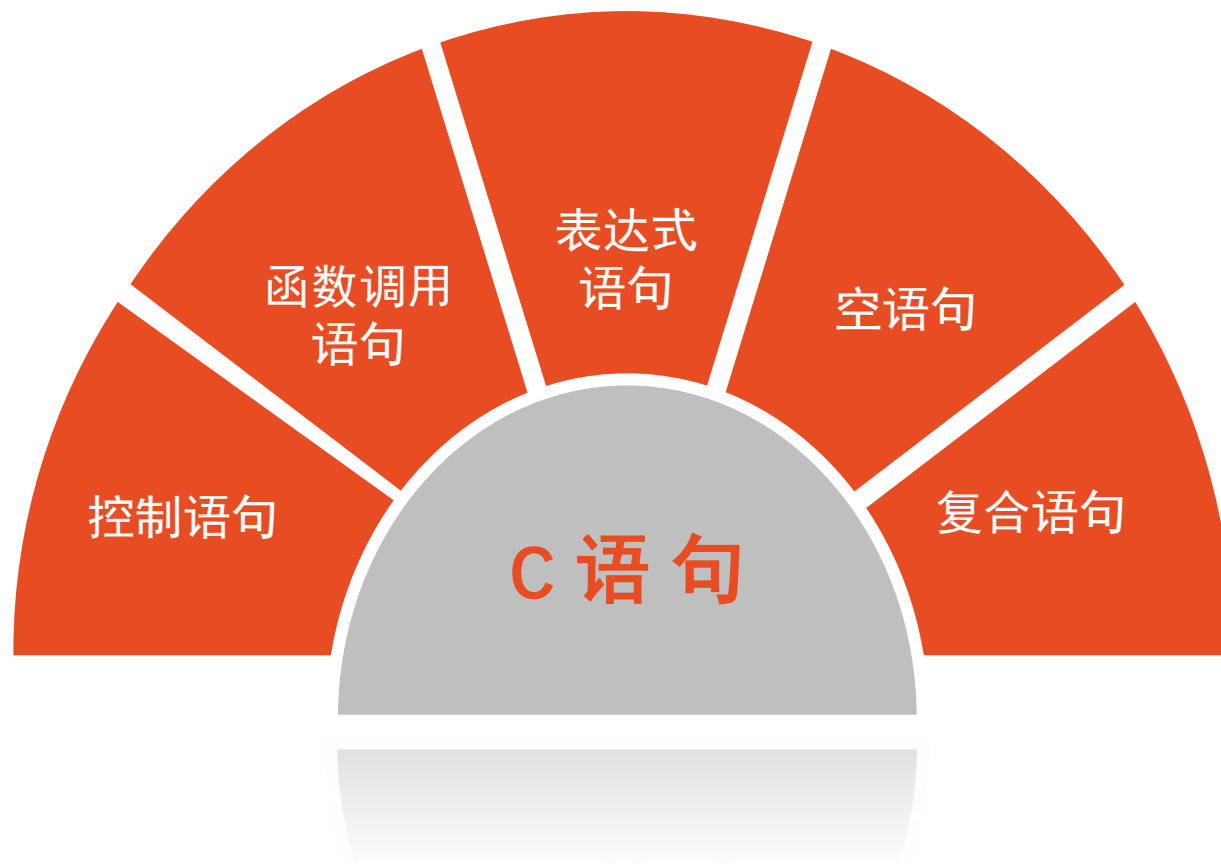
(int)x+y 只将x转换成整型，然后与y相加

```
int a; float x;
```

```
a=(int)x;
```

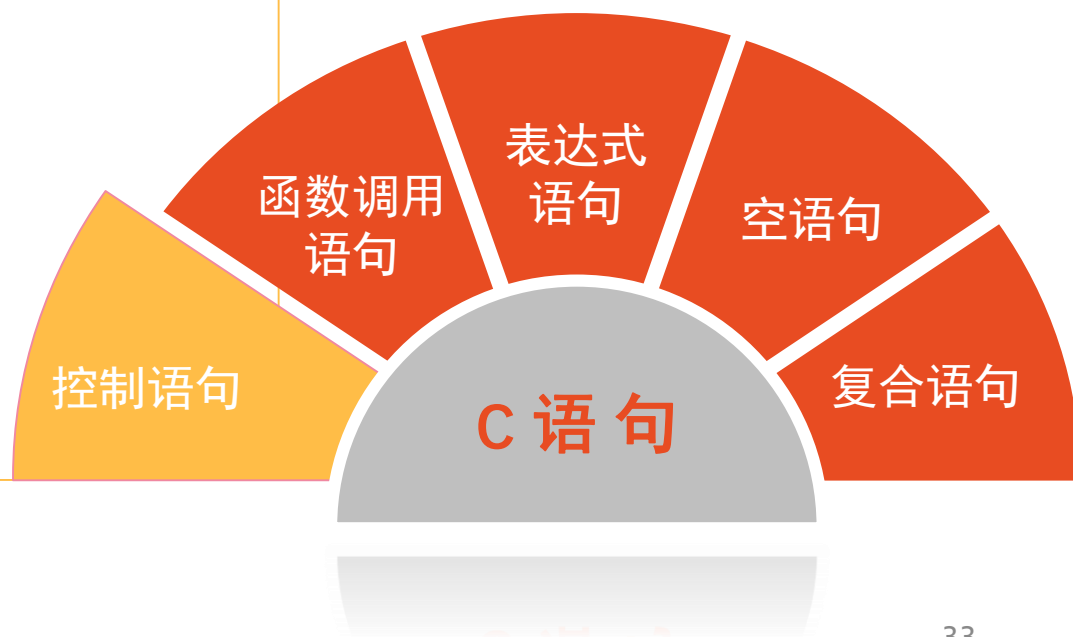
进行强制类型运算(int)x后得到一个int类型的临时值，它的值等于 x 的整数部分，把它赋给a，注意x的值和类型都未变化，仍为float型。该临时值在赋值后就不再存在了。

C 语 句



- ① if()…else…（条件语句）
- ② for()…（循环语句）
- ③ while()…（循环语句）
- ④ do…while ()（循环语句）
- ⑤ continue（结束本次循环语句）
- ⑥ break（中止执行switch或循环语句）
- ⑦ switch（多分支选择语句）
- ⑧ return（从函数返回语句）

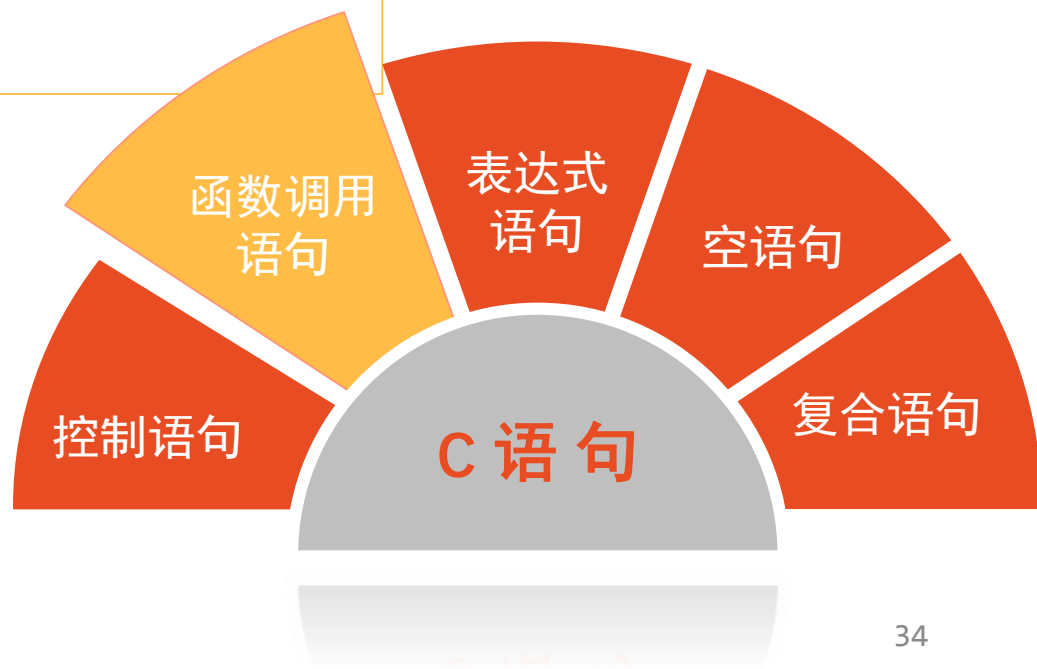
()表示括号中是一个判别条件
…表示内嵌的语句



函数调用语句由一个函数调用加一个分号构成。

```
printf("This is a C statement. ");
```

其中printf("This is a C statement. ")是一个函数调用，加一个分号成为一个语句。



表达式语句由一个表达式加一个分号构成，
最典型的是由赋值表达式构成一个赋值语句。

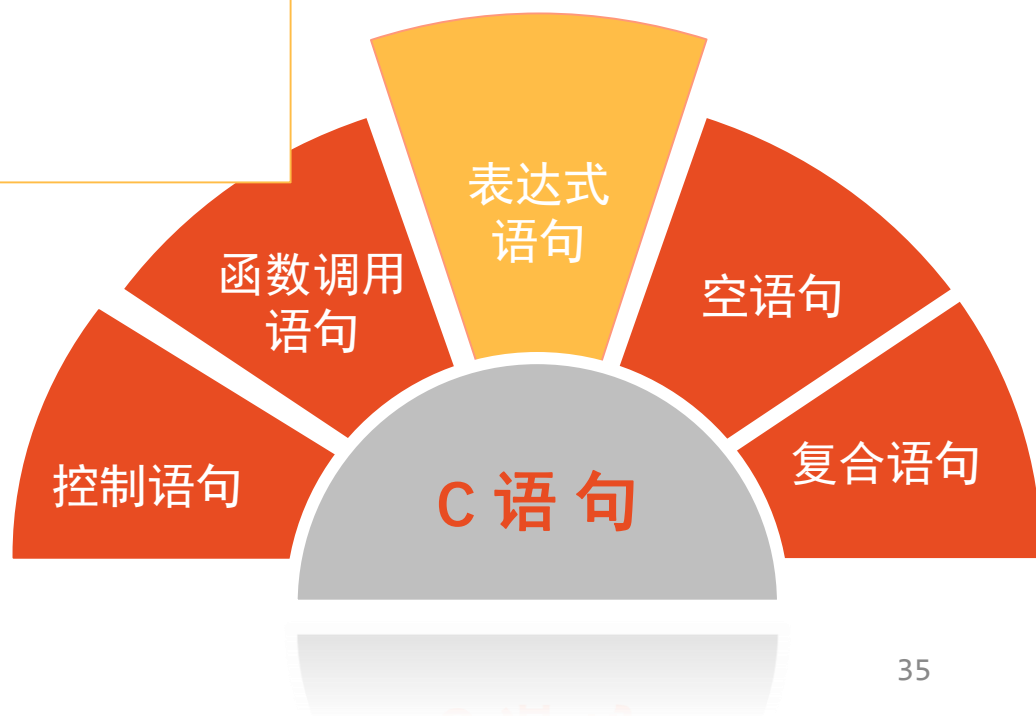
例如：

`a=3`

是一个赋值表达式，而

`a=3;`

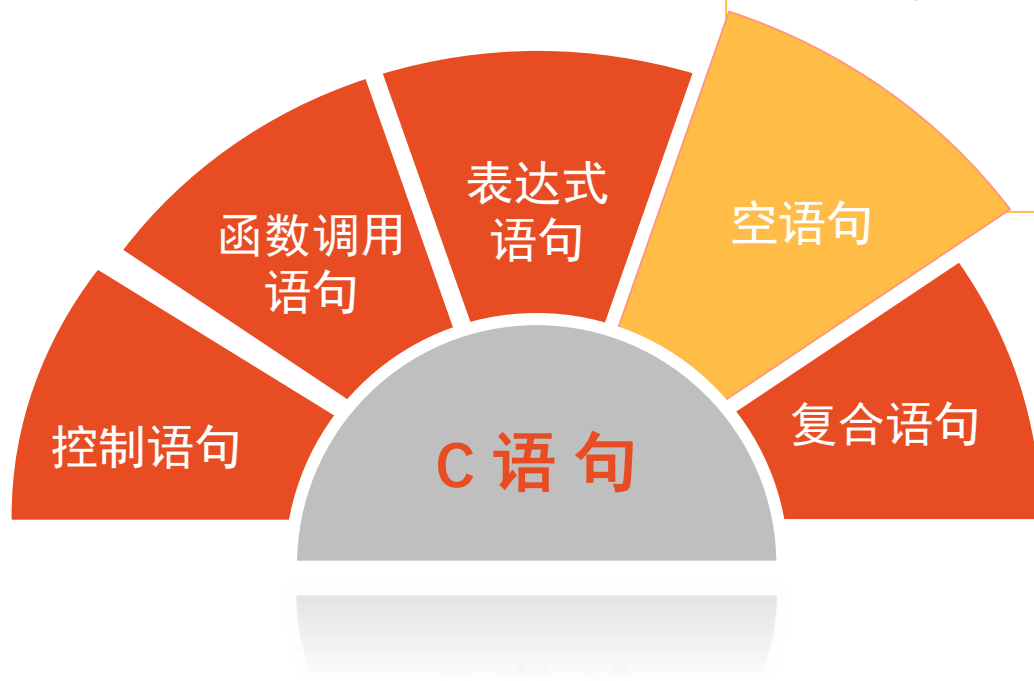
是一个赋值语句。



;

只有一个分号的语句即为空语句。

可用来作为循环语句中的循环体（循环体是空语句，表示循环体什么也不做）。



可以用{}把一些语句和声明括起来成为复合语句(又称语句块)。

```
if(a > b)
{
    t = a;
    a = b;
    b = t;
}
```

复合语句常用在if语句或循环中，此时程序需要连续执行一组语句。



赋值运算符 “=”

“=” 的作用是将一个数据赋给一个变量。

例如：a=3的作用是执行一次赋值操作（或称赋值运算）。

把常量3赋给变量a。

也可以将一个表达式的值赋给一个变量。

赋值表达式：变量 赋值运算符 表达式

赋值表达式的作用是将一个表达式的值赋给一个变量，因此赋值表达式具有计算和赋值的双重功能。

对赋值表达式求解的过程是：

①求赋值运算符右侧的“表达式”的值，②赋给赋值运算符左侧的变量。既然是一个表达式，就应该有一个值，赋值表达式的值等于赋值后左侧变量的值。

变量赋初值

可以用赋值语句对变量赋值，也可以在定义变量时对变量赋以初值。

```
int a=3;           //指定a为整型变量，初值为3；  
                  //相当于int a; a=3;  
float f=3.56;      //指定f为浮点型变量，初值为3.56  
char c='a';        //指定c为字符变量，初值为'a'  
int a,b,c=5;       //指定a, b ,c为整型变量，但只对c初始  
                  //化，c的初值为5；  
                  //相当于int a,b,c; c=5;
```


赋值语句

【例6.2】给出三角形的三边长，求三角形面积。

解题思路：假设给定的三个边符合构成三角形的条件：任意两边之和大于第三边。

从数学知识已知求三角形面积的公式为： $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$ ，其中 $s = (a+b+c)/2$ 。

赋值语句

【例6.2】 给出三角形的三边长，求三角形面积。

```
1 #include <stdio.h>
2 #include <math.h>
3 int main ()
4 {   double a,b,c,s,area;//定义各变量，均为double型
5     a=3.67;      //对边长a赋值
6     b=5.43;      //对边长b赋值
7     c=6.21;      //对边长c赋值
8     s=(a+b+c)/2; //计算s
9     area=sqrt(s*(s-a)*(s-b)*(s-c)); //计算area
10    printf("a=%f\tb=%f\tc=%f\n",a,b,c); //输出三边a,b,c的值
11    printf("area=%f\n",area); //输出面积area的值
12    return 0;
13 }
```



为提高精度，变量都定义为双精度类型



sqrt函数是求平方根的函数。由于要调用数学函数库中的函数，必须在程序的开头加一条#include指令，把头文件“math.h”包含到程序中来。



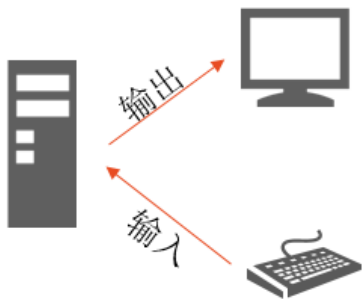
转移字符'\t'用来调整输出的位置，使输出的数据清晰、整齐、美观

```
C:\WINDOWS\system32\cmd.exe
a=3.670000    b=5.430000    6.210000
area=9.903431
请按任意键继续. . .
```

数据的输入输出

有关输入输出的概念

- 1 输入输出是以计算机主机为主体而言的



- 2 C语言本身不提供输入输出语句

输入和输出操作是由C标准函数库中的函数来实现的。

优点：

简化编译系统

增强通用性和可移植性

- 3 要在程序文件的开头用预处理指令#include把有关头文件放在本程序中

```
#include<stdio.h>
```

printf函数

用来向终端（或系统隐含指定的输出设备）输出若干个任意类型的数据。

printf（格式控制，输出列表）

```
printf("i=%d,c=%c\n", i, c )
```

普通字符 格式声明 输出列表

(1) “**格式控制**”是用双引号括起来的一个字符串，称为格式控制字符串，简称格式字符串。包括：

①**普通字符**。普通字符即需要在输出时原样输出的字符。

②**格式声明**。格式声明由“%”和格式字符组成。作用是将输出的数据转换为指定的格式后输出。

(2) **输出列表**是程序需要输出的一些数据，可以是常量、变量或表达式。

printf函数——格式声明

整型数据	%d	以有符号十进制形式输入/输出整型数
实型数据	%f	以小数形式输入/输出实型数
字符型数据	%c	输入/输出一个字符
	%s	输入/输出字符串

scanf函数 用来输入数据。

scanf（格式控制，地址表列）

```
scanf("%f%f%f", &a, &b, &c);
```

格式声明

地址列表

(1) “格式控制” 是用双引号括起来的一个字符串，含义同printf函数。包括：

- ① 普通字符。 输入整型或者浮点类型不要加任何普通字符！
- ② 格式声明。以%开始，以一个格式字符结束，中间可以插入附加的字符。

(2)地址表列是由若干个地址组成的表列

【例6.3】求 $ax^2+bx+c=0$ 方程的根。a,b,c由键盘输入，设 $b^2-4ac > 0$ 。

解题思路:首先要知道求方程式的根的方法。由数学知识已知: 如果 $b^2-4ac \geq 0$, 则一元二次方程有两个实根: $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$, $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$, 将分式分为两项: $p = \frac{-b}{2a}$, $q = \frac{\sqrt{b^2-4ac}}{2a}$, 则 $x_1=p+q$, $x_2=p-q$, 有了这些式子, 只要知道a,b,c的值, 就能顺利地求出方程的两个根。

```
1 #include <stdio.h>
2 #include<math.h> //程序中要调用求平方根函数sqrt
3 int main()
4 {   double a,b,c,disc,x1,x2,p,q; //disc用来存放判别式(b*b-4ac)的值
5     scanf("%lf%lf%lf",&a,&b,&c); //输入双精度型要用格式声明"%lf"
6     disc=b*b-4*a*c;
7     p=-b/(2.0*a);
8     q=sqrt(disc)/(2.0*a);
9     x1=p+q;
10    x2=p-q; //求出方程的两个根
11    printf("x1=%7.2f\nx2=%7.2f\n",x1,x2); //输出方程的两个根
12    return 0;
13 }
```


【例6.3】求 $ax^2+bx+c=0$ 方程的根。a,b,c由键盘输入，设 $b^2-4ac > 0$ 。

```
1 #include <stdio.h>
2 #include<math.h> //程序中要调用求平方根函数sqrt
3 int main()
4 {   double a,b,c,disc,x1,x2,p,q; //disc用来存放判别式(b*b-4ac)的值
5     scanf("%lf%lf%lf",&a,&b,&c); //输入双精度型要用格式声明"%lf"
6     disc=b*b-4*a*c;
7     p=-b/(2.0*a);
8     q=sqrt(disc)/(2.0*a);
9     x1=p+q;
10    x2=p-q; //求出方程的两个根
11    printf("x1=%7.2f\nx2=%7.2f\n",x1,x2); //输出方程的两个根
12    return 0;
13 }
```



scanf函数用于输入a,b,c的值。

函数中括号内变量a,b,c的前面，要用地址符&。&a表示变量a在内存中的地址。

双引号内用%lf格式声明，表示输入的是双精度型实数。

格式声明为“%lf%lf%lf”，要求输入3个双精度实数。程序运行时，输入“1 3 2”，两个数之间用空格分开。输入的虽是整数，但由于指定用%lf格式输入，因此系统会先把这3个整数转换成实数1.0,3.0,2.0，然后赋给变量a,b,c。

【例6.3】求 $ax^2+bx+c=0$ 方程的根。a,b,c由键盘输入，设 $b^2-4ac > 0$ 。

```
1 #include <stdio.h>
2 #include<math.h> //程序中要调用求平方根函数sqrt
3 int main()
4 {   double a,b,c,disc,x1,x2,p,q; //disc用来存放判别式(b*b-4ac)的值
5     scanf("%lf%lf%lf",&a,&b,&c); //输入双精度型要用格式声明"%lf"
6     disc=b*b-4*a*c;
7     p=-b/(2.0*a);
8     q=sqrt(disc)/(2.0*a);
9     x1=p+q;
10    x2=p-q; //求出方程的两个根
11    printf("x1=%7.2f\nx2=%7.2f\n",x1,x2); //输出方程的两个根
12    return 0;
13 }
```

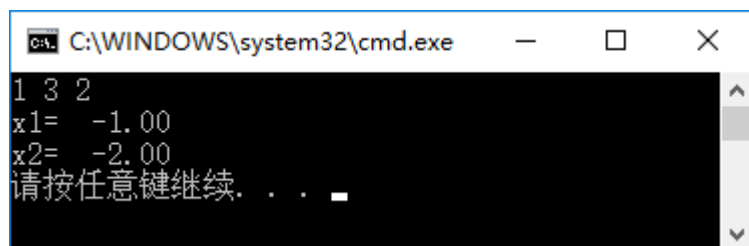


在printf函数中，在格式符f的前面加了“7.2”，表示在输出x1和x2时，指定数据占7列，其中小数占2列。优点：

- ①可以根据实际需要来输出小数的位数；
- ②如果输出多个数据，可使输出数据整齐美观。

【例6.3】求 $ax^2+bx+c=0$ 方程的根。a,b,c由键盘输入，设 $b^2-4ac > 0$ 。

```
1 #include <stdio.h>
2 #include<math.h> //程序中要调用求平方根函数sqrt
3 int main()
4 {   double a,b,c,disc,x1,x2,p,q; //disc用来存放判别式(b*b-4ac)的值
5     scanf("%lf%lf%lf",&a,&b,&c); //输入双精度型要用格式声明"%lf"
6     disc=b*b-4*a*c;
7     p=-b/(2.0*a);
8     q=sqrt(disc)/(2.0*a);
9     x1=p+q;
10    x2=p-q; //求出方程的两个根
11    printf("x1=%7.2f\nx2=%7.2f\n",x1,x2); //输出方程的两个根
12    return 0;
13 }
```



```
C:\WINDOWS\system32\cmd.exe
1 3 2
x1= -1.00
x2= -2.00
请按任意键继续. . . .
```



putchar函数

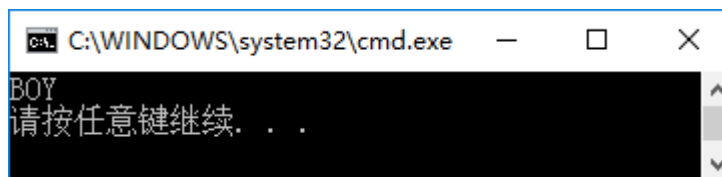
从计算机向显示器输出一个字符。

putchar(c)

【例6.4】先后输出BOY三个字符。

解题思路：定义3个字符变量，分别赋以初值'B'，'O'，'Y'，然后用putchar函数输出这3个字符变量的值。

```
1 #include <stdio.h>
2 int main()
3 {   char a='B',b='O',c='Y';//定义3个字符变量并初始化
4     putchar(a);//向显示器输出字符B
5     putchar(b);//向显示器输出字符O
6     putchar(c);//向显示器输出字符Y
7     putchar ('\n');//向显示器输出一个换行符
8     return 0;
9 }
```



getchar函数

向计算机输入一个字符。

`getchar()`

函数没有参数。

函数的值就是从输入设备得到的字符。

只能接收一个字符。

如果想输入多个字符就要用多个函数。

不仅可以从输入设备获得一个可显示的字符，而且可以获得控制字符。

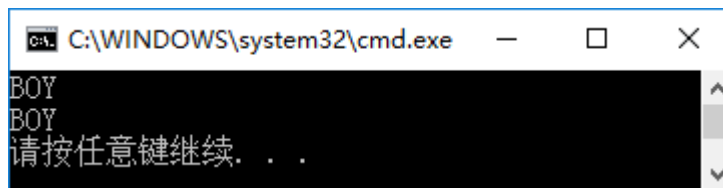
用getchar函数得到的字符可以赋给一个字符变量或整型变量，也可以作为表达式的一部分。如，`putchar(getchar());`将接收到的字符输出。

getchar函数

【例6.5】从键盘输入BOY 3个字符，然后把它们输出到屏幕。

解题思路：用3个getchar函数先后从键盘向计算机输入BOY 3个字符，然后用putchar函数输出。

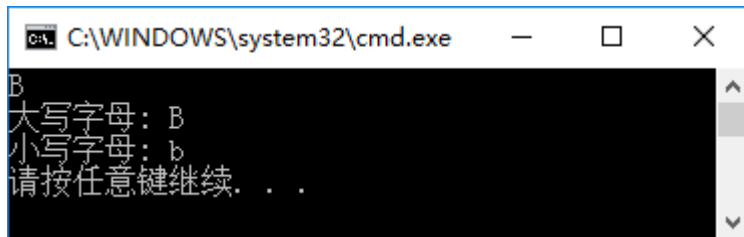
```
1 #include <stdio.h>
2 int main()
3 {   char a,b,c; //定义字符变量a,b,c
4     a=getchar(); //从键盘输入一个字符，送给字符变量a
5     b=getchar(); //从键盘输入一个字符，送给字符变量b
6     c=getchar(); //从键盘输入一个字符，送给字符变量c
7     putchar(a); //将变量a的值输出
8     putchar(b); //将变量b的值输出
9     putchar(c); //将变量c的值输出
10    putchar('\n'); //换行
11    return 0;
12 }
```



【例6.6】从键盘输入一个大写字母，在显示屏上显示对应的小写字母。

解题思路：用getchar函数从键盘读入一个大写字母，把它转换为小写字母，然后用putchar或者printf函数输出该小写字母。

```
1 #include <stdio.h>
2 int main ()
3 {   char c1,c2;
4     c1=getchar();//从键盘读入一个大写字母，赋给字符变量c1
5     c2=c1+32;    //得到对应的小写字母的ASCII代码，放在字符变量c2中
6     printf("大写字母: %c\n小写字母: %c\n",c1,c2);//输出c1,c2的值
7     return 0;
8 }
```



```
C:\WINDOWS\system32\cmd.exe
B
大写字母: B
小写字母: b
请按任意键继续. . .
```


课后练习

1. 登录在线评测系统，做编程练习。