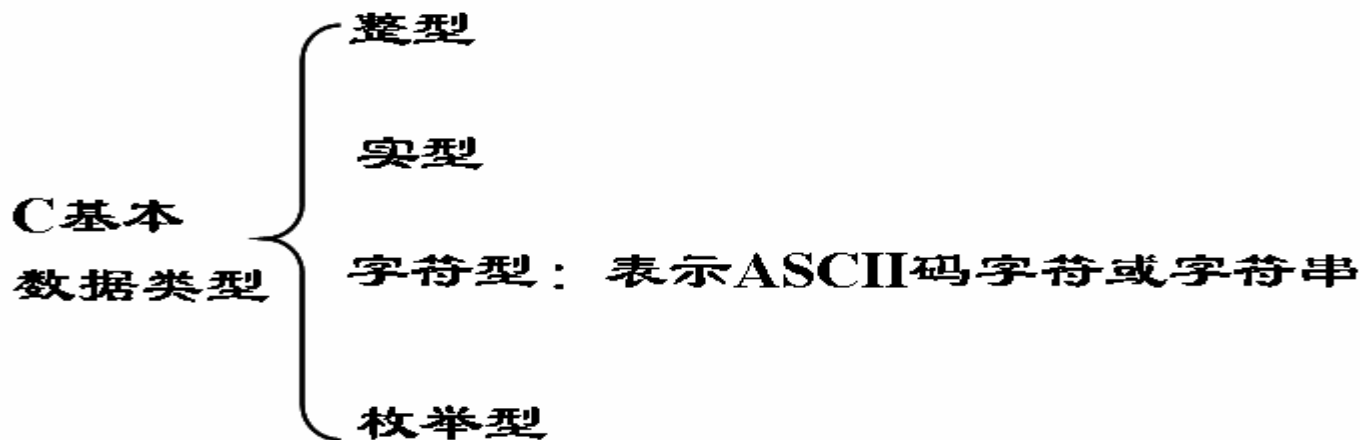


第3章 数据类型、运算符及表达式





§ 1 基本数据类型一览

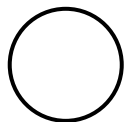


American Standard Code for Information Interchange——ASCII码，美国信息交换标准码。

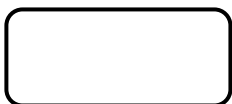
- 1.取值：**ASCII码取值范围为0~255，占据1个字节
- 2.分类：**分为普通（可显示）字符和控制（不可显示）两类
- 3.数字字符：**'0'+1='1'; '1'+1='2'.....'8'+1='9'
- 4.字母字符：**'a'-'A'='b'-'B'=.....='z'-'Z'=32



§ 2 语法图



表示一个ASCII码字符



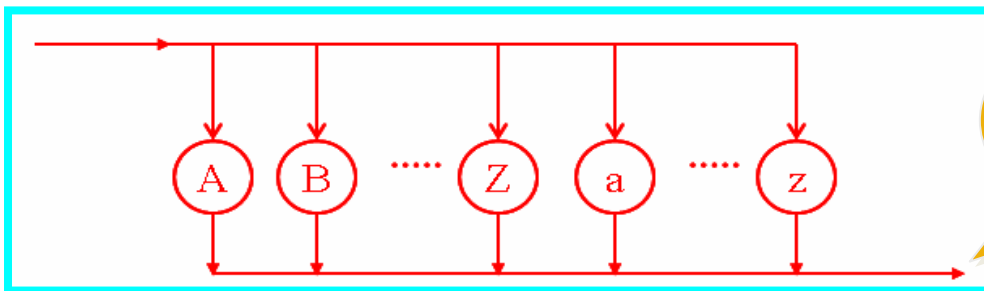
表示关键字



表示另一个语法图



指明语法图中某条路径的合法方向



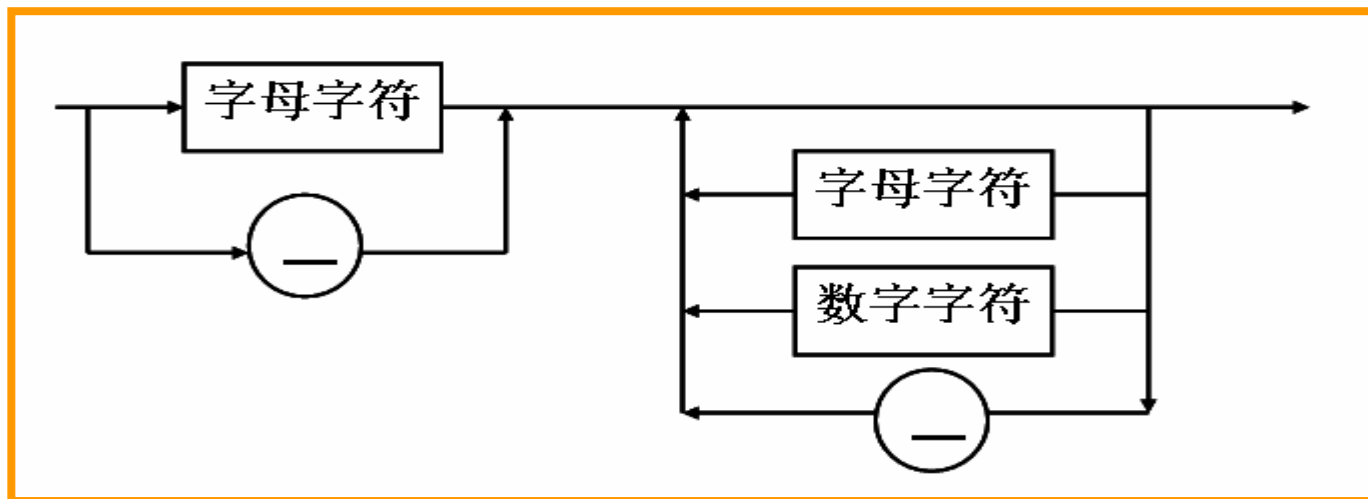
字母字符
语法

语法图含义：一个语法图中，从箭头入口到出口全部可能的路径集即构成该语法的所有合法形式。



§ 3 标识符

标识符是一个名字，是在程序中用来标识变量名、符号常量名、函数名、数组名等的有效字符序列



eg: a、_、sum、s1、_above、day_of_year

合法

eg: d.m、a\$、x1-1、2y

非法



§ 4 常量与变量

一、符号常量

例1:编程输出半径为r的圆周长、圆面积及球体积之值

解:

```
main()  
{ float r, l, s, v;  
  scanf("%f", &r);  
  l=2.0*3.14*r;  
  s=3.14*r*r;  
  v=4.0/3.0*3.14*r*r*r;  
  printf("%f,%f,%f", l, s, v);  
}
```



§ 4 常量与变量

一、符号常量

- 符号常量常是醒目的大写标识符，用以表示特定的常量串
- **宏定义命令**：**#define** 符号常量名 常量串

宏定义 → 宏调用 → 宏展开

I解：

```
main()  
{ float r, l, s, v;  
  scanf("%f", &r);  
  l=2.0*3.14*r;  
  s=3.14*r*r;  
  v=4.0/3.0*3.14*r*r*r;  
  printf("%f,%f,%f", l, s, v);  
}
```

II解：

```
#define PI 3.14
```

```
main()  
{ float r, l, s, v;  
  scanf("%f", &r);  
  l=2.0*PI*r;  
  s=PI *r*r;  
  v=4.0/3.0*PI*r*r*r;  
  printf("%f,%f,%f", l, s, v);}
```



§ 4 常量与变量

一、符号常量

例2：分析下列程序的运行结果

```
#define X 5
#define Y X+1
main( )
{ int a;
  a=Y;
  printf(“%d\n”,X);
  printf(“%d\n”, a+1);
}
```

宏展开后的程序面貌：

```
main( )
{ int a;
  a=5+1;
  printf(“%d\n”,5);
  printf(“%d\n”, a+1);
}
```

5

7



§ 4 常量与变量

一、符号常量

例3:

(1) 设有宏定义:

```
#define WIDTH 80
```

```
#define LENGTH WIDTH+40
```

则执行语句 $v=LENGTH*20$ 后, v 的值是 880。

(2) 设有宏定义:

```
#define WIDTH 80
```

```
#define LENGTH (WIDTH+40)
```

则执行语句 $v=LENGTH*20$ 后, v 的值是 2400。



§ 4 常量与变量

二、变量

有唯一的变量名:变量名取名使用标识符规则

变量占据存储单元的长度取决于变量的类型

C语言规定: 变量必须先定义, 再使用

变量类型定义语句: 类型说明符 变量名[,变量名]... ;

变量取名应见名知意: average sum year 等

eg1: int number; /*number是整型变量*/

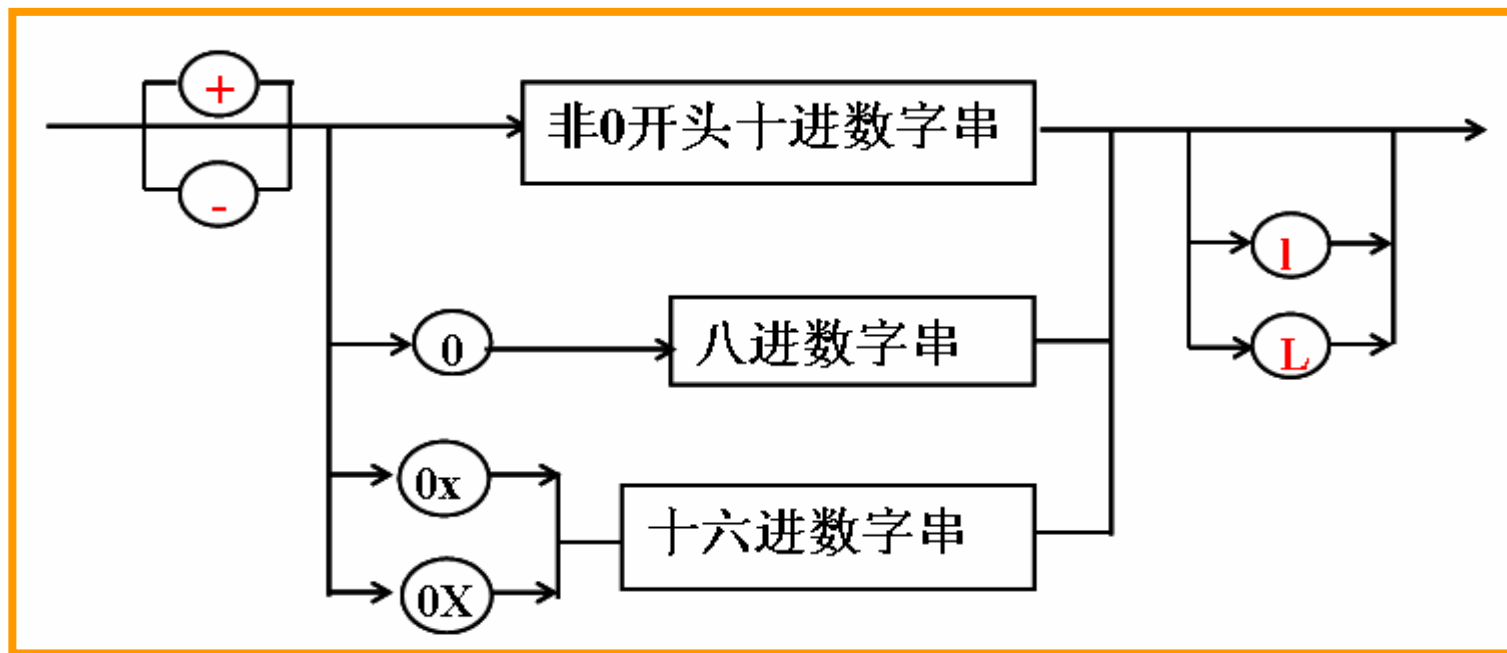
eg2: char sex; /*sex是字符型变量*/

eg3: float aver,sum; /*实型变量定义*/



§ 5 整型数据

一、整型常量



1 C程序中,整型常量可用十、八、十六进制加以表示

2 整型常量可用语法后缀L(l), 表示是长整型数

§ 5 整型数据

二、整型变量



eg1: signed int i;

等价

eg1: int i;

eg2: signed short int h;

eg2: short int h;

eg2: signed short h;

eg2: short h;

§ 5 整型数据

二、整型变量



```
eg3: signed long int l;
```

```
eg3: long int l;
```

```
eg3: signed long l;
```

```
eg3: long l;
```

§ 5 整型数据

二、整型变量



eg4: unsigned int u;

eg4: unsigned u;

eg5: unsigned long int lu;

eg5: unsigned long lu;

§ 5 整型数据

二、整型变量



防止
溢出!

整数类型	字节数(位数)	取值范围
int	2(16bit)	-32768~32767: $-2^{15} \sim 2^{15}-1$
short int	2(16bit)	-32768~32767: $-2^{15} \sim 2^{15}-1$
long int	4(32bit)	-2147483648~2147483647: $-2^{31} \sim 2^{31}-1$
unsigned int	2(16bit)	0~65535: $0 \sim 2^{16}-1$
unsigned short	2(16bit)	0~65535: $0 \sim 2^{16}-1$
unsigned long	4(32bit)	0~4294967295: $0 \sim 2^{32}-1$

§ 5 整型数据

例1：写出下列程序的运行结果。

```
main( )  
{ int a,b;  
  a=32767;  
  b=a+1;  
  printf("a=%d,b=%d\n",a,b);  
}
```

a=32767,b=-32768

a

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

b

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(-32768)



§ 5 整型数据

例2:写出下列程序的运行结果

```
main( )  
{ int i;  
  i=50000;  
  printf("i=%d\n",i);  
}
```

```
main( )  
{ unsigned i;  
  i=50000;  
  printf("i=%u\n",i);  
}
```

50000

1 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0

(-15536)补

i=-15535

i=50000

§ 6 实型数据

一、实型常量

1.

实数在C程序中一律用十进制表示

2.

十进制小数:由十进数字、小数点和数符构成, 且小数点不可省略

3.

十进制指数:由尾数、E或e以及指数构成: **xen**或**xEn**

—— xen表数值 $x \times 10^n$

—— x可以是十进制整数或小数, n必须是十进制整数

eg1: 9.6 -3.5 .9 -5. 5.

eg2: 5e8 5.1e-9 5.e1 -.9e-2

eg3: e8 (1e8) 5e (5e0) 5.e1.2 (×)



§ 6 实型数据

二、实型变量

- **单精度型float** : 占4个字节, 提供7位有效数字
- **双精度型double** : 占8个字节, 提供16位有效数字

```
eg: float  a,b;  
    a=12345.6789;  
    b=.0000000026;  
    :
```

内存值:

```
a=12345.6700;  
b=.00000000;
```

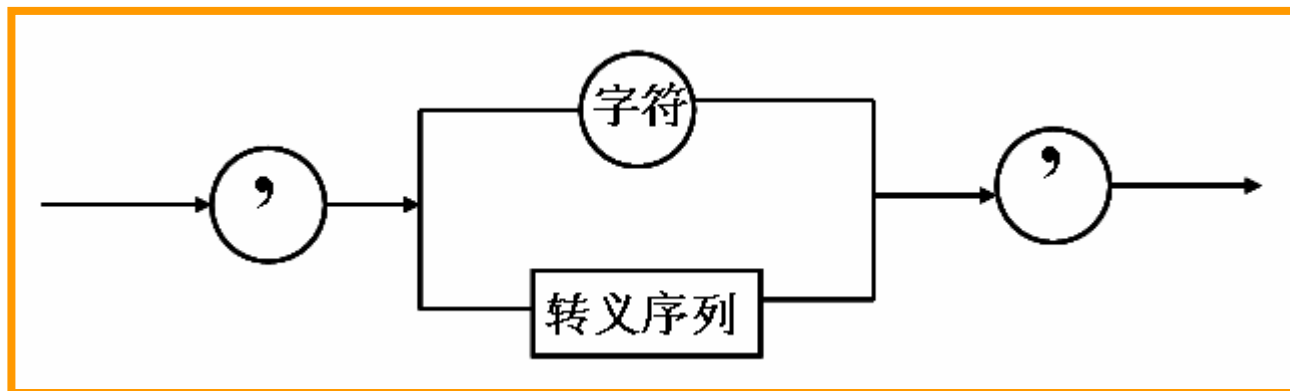
```
eg: double a,b;  
    a=12345.6789;  
    b=.0000000026;  
    :
```

```
a=12345.6789.....;  
b=.0000000026.....;
```



§ 7 字符型数据

一、字符常量



转义 序列

由倒斜杠后跟特定字符组成，将后继字符转变为另一种解释，形成转义

eg: '0' 'a' 'A'

eg: '\b' '\n'



§ 7 字符型数据

一、字符常量

转义序列	含义
<code>\n</code>	回车换行
<code>\t</code>	横向制表
<code>\b</code>	退格
<code>\r</code>	只回车不换行
<code>\\</code>	倒斜杠字符\
<code>\'</code>	单引号字符'
<code>\"</code>	双引号字符"
<code>\0</code>	空字符NULL
<code>\ddd</code>	1-3位8进制数表示的字符
<code>\xhh</code>	1-2位16进制数表示的

eg1: `printf("AB");`

eg2: `printf("A\nB");`

eg3: `printf("A\tB");`

eg4: `printf("sssss\b\b");`

eg5: `printf("ssss\b123");`

eg6: `printf("aaaaa\rbb\n");`

eg7: `printf("YYY\r===");`

eg8: `printf("aaaaa\bx\rbb\tcd\n");`



§ 7 字符型数据

一、字符常量

转义序列	含义
<code>\n</code>	回车换行
<code>\t</code>	横向制表
<code>\b</code>	退格
<code>\r</code>	只回车不换行
<code>\\</code>	倒斜杠字符\
<code>\'</code>	单引号字符'
<code>\''</code>	双引号字符"
<code>\0</code>	空字符NULL
<code>\ddd</code>	1-3位8进制数表示的字符
<code>\xhh</code>	1-2位16进制数表示的字符

eg9: `printf("\");` /***语法出错***/

eg9: `printf("\\");` /***正确***/

eg10: `''` → `\'`

eg11: `printf("");` /***不提倡***/

eg11: `printf("\'");`

eg12: `printf("A");`
`printf("\\101");`
`printf("\\x41");`
/***均为输出A***/



§ 7 字符型数据

一、字符常量

eg1: 说出以下数中哪些是合法的数 ?

'\3' '\12' '\19' '\132' '\1234'
'\777' '\8' '\07' '07' 07

eg2: 说出以下数中哪些是合法的数 ?

'\xf' '\x1a' '0x1a' 0x1a
0xcl '\xcl' '\x111' '\xff'



§ 7 字符型数据

二、字符变量

● **定义语句：**char 变量名[, 变量名]...;

— 占用1个字节的内存，以ASCII值加以存放

— 字符型数据与整型数据可在一定范围等价通用

— 字符型数据间可进行加、减、比较运算,是ASCII值的运算

例1: `printf("%d,%c",'a'-'A','a'-'A');`

例2: `printf("%d,%c",'0'+'A','0'+'A');`

例3: 若变量c中是小写字母，则将其转变为对应的大写字母

```
if(c>='a' && c<='z') c=c-('a'-'A');
```

例4: 若变量c是数字字符，将其转化为对应的数字。

```
if(c>='0' && c<='9') c=c-'0';
```



§ 7 字符型数据

三、字符串常量

● **字符串常量是指由双引号对括的字符和转义序列集合**

—— C语言中无字符串型变量

—— 在C语言中，字符串以字符类型的数组加以存放

—— 在存放字符串时，系统自动加上串结束标志\0

“BASIC”/*长度为5的串*/

“Don\'t move!” /*长度为11的串*/

“No.\tname\tsex\tage\n” /*长度为17的串*/

printf(“No.\tname\tsex\tage\n”);

“” /*长度为0的空串*/

“I_ am_ a\0 student.” /*串长为6，表示I am a*/



§ 7 字符型数据

题1：在C语言中，十进制47可等价地写为_____。

- A. 2f B. 02f C. 57 D. 057

题2：在C语言中，错误的int型常量是_____。

- A. 32768 B. 0 C. 037 D. 0xAF

题3：下列可以正确表示字符型常量的是_____。

- A. "a" B. '\\ ' C. '\081' D. a



§ 7 字符型数据

题4：下列字符常量中，不合法的是_____。

- A. '\0xff' B. '\65' C. '\$' D. '\x1a'

题5：若有说明语句：char c='\72'; 则变量c_____。

- A. 包含1个字符 B. 包含2个字符
C. 包含3个字符 D. 说明不合法，c的值不定

题6：字符串"\x69\082\n"的长度是_____。

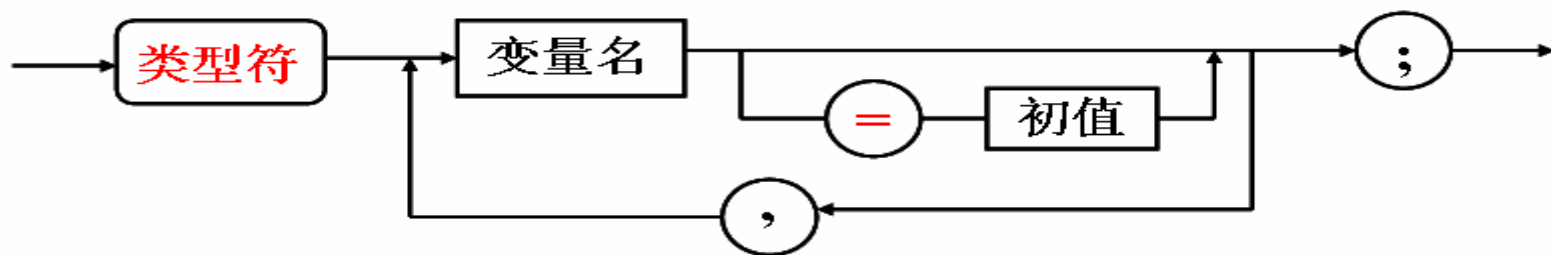
- A. 3 B. 5 C. 1 D. 6

题7：字符串"\x69\02\n"的长度是_____。

- A. 3 B. 5 C. 1 D. 6



§ 8 变量定义及初始化



eg1: **int** a,b;
short c,d;
long l;
unsigned u1,u2;
unsigned short h1,h2;
unsigned long l1,l2;
float e,f,g;
double d1;
char c1,c2,c3;

- 变量定义语句是非执行语句
- 允许变量初始化,允许对部分变量初始化
- 不允许使用=号连续初始化

eg2: **float** f1=3.5; /***float** f1;f1=3.5*/
double d1;

eg3: **float** a,b=-9.7,c; /***正确***/

eg4: **int** a=b=c=2; /***语法错误***/

eg4: **int** a=2,b=2,c=2; /***正确***/



§ 9 算术运算

一、算符及语义

+ **-** ***** **/** **%**

—— **%**表示取余(求模)运算，只适用于整型数据

—— **/**用于整型数据时，表示整

—— 算符的**优先级**:指**不同级别**算符运算的先后次序

—— 算符的**结合性**:指**同级**算符运算的先后次序

eg1: printf(“%d,%d\n”, 5%2,4%2); **/*1,0 */**

eg2: printf(“%d,%d,%d\n”, -5%2,5%-2,-9%-5); **/*-1,1,-4 */**

eg3: printf(“%f,%f,%d\n”,5./2,5/2.,5/2); **/*2.5,2.5,2*/**

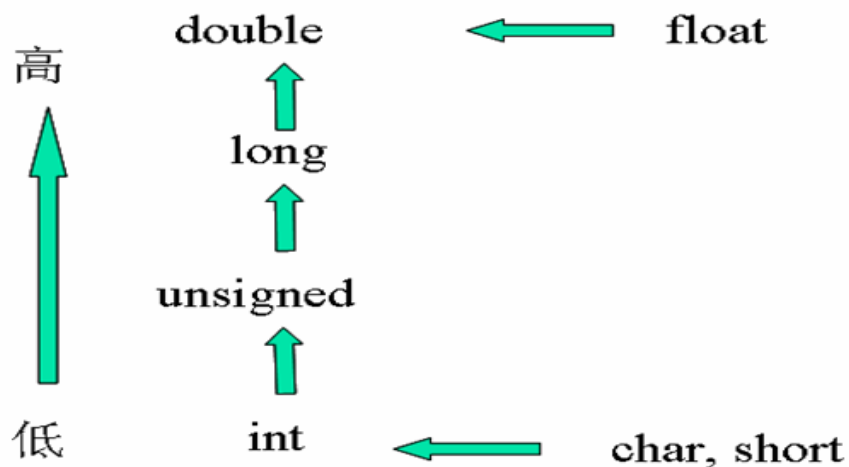


§ 9 算术运算

二、运算数据的类型转换

自动隐式转换

- 由编译系统自动进行
- 横向为必然转换
- 纵向为混合运算时的升格规则



eg1: 'a'+'b' /*结果类型int*/

eg2: 1.5-2.6 /*double型*/

eg3: 'a'+5-87.9 /*double型*/

eg4:有 int i;
float f;
double d;
long e;
分析: $10+'a'+i*f-d/e$

eg5:有 int i,j; float x;
求: $i = (\text{int}) x \% j;$



§ 9 算术运算

二、运算数据的类型转换

自动隐式转换

- 由编译系统自动进行
- 纵向为必然转换
- 纵向为混合运算时的升格规则



强制显式转换

- 强制类型运算符: **(类型)**
- 强制类型表达式: **(类型)(表达式)**
将表达式结果转换成类型名所指类型
- 强制类型转换只改变变量参加运算之值,其内存的值及类型不变

eg6: float x=1.2,y=2.3,p=2.5;

```
printf(“%d”,(int)(x+y)%(int)p); /*输出1*/
```

eg7: float x=3.6,y=4.7;

```
i=(int)x+(int)y; /*i=7 */
```

```
z=x+y; /*z=8.3*/
```



§ 10 赋值运算符=

一、赋值表达式

变量**=**表达式

计算左值表达式之值，送入左值变量内存单元中

在C语言中，赋值表达式之值等于左值变量之值

eg1: `c='x'+1`

eg2: `a=x*y%z-x`

eg3: `printf("%d",a=3);`

赋值号
与等于号

等于号具有恒等性，赋值号具有不等性

等于号具有互易性，赋值号具有方向性



§ 10 赋值运算符=

一、赋值表达式

变量=表达式

计算左值表达式之值，送入左值变量内存单元中

在C语言中，赋值表达式之值等于左值变量之值

eg1: `c='x'+1`

eg2: `a=x*y%z-x`

eg3: `printf("%d",a=3);`

eg4: `a=(b=10)/(c=3)` /*a为3, b为10, c为3*/

eg5: `a=b=c=5`

等价

`a=(b=(c=5))`





§ 10 赋值运算符=

二、赋值中的类型匹配

原则：总是将表达式结果的类型转换为变量所需类型后，再进行赋值

例1：int i=23;
float f;
f=i; /*f=23.0*/

升

例2：float f=3.56;
int i;
i=f; /*i=3*/

降

例3：
main()
{ char c;
int i;
c='\376';
i=c;
printf("i=%d\n",i);
}

i=-2

例4：
main()
{ unsigned char c;
int i;
c='\376';
i=c;
printf("i=%d\n",i);
}

i=254



§ 10 赋值运算符=

二、赋值中的类型匹配

原则：总是将表达式结果的类型转换为变量所需类型后，再进行赋值

例5：

```
main( )  
{ unsigned a=65535;  
  int b;  
  b=a;  
  printf("b=%d\n",b);  
}
```

b=-1

例6：

```
main( )  
{ unsigned a;  
  int b=-1;  
  a=b;  
  printf("a=%u\n",a);  
}
```

a=65535



§ 10 赋值运算符=

三、复合赋值运算符

变量 **OP=** 表达式 \longrightarrow 变量 = 变量**OP**表达式

——操作符OP是二目算术运算符或是二目位运算

eg: $a+=3;$	等价于	$a=a+3;$
$x\%=(y+3);$	等价于	$x=x\%(y+3);$
$x-=1;$	等价于	$x=x-1;$

例1：写出下列程序的运行结果

```
main()  
{ int n=3;  
  printf(“%d,%d”, n ,1+3*(n+=5)/5); /*8,5*/  
}
```

例2：

已知int x=6;执行
 $x+=x-=x*x;$ 后,
x之值是 -60。



§ 11 增、减1算符

$n=n+1$

$n+=1$

$++\ n$

$n++$

$n=n-1$

$n-=1$

$--\ n$

$n--$

**$++$ 是增1
运算符；**

**$--$ 是减1
运算符；**



§ 11 增、减1算符

- ++ 用作**前缀**,表示**先增1**再引用; 用作**后缀**,表示**先引用**再增1
- -- 用作**前缀**,表示**先减1**再引用; 用作**后缀**,表示**先引用**再减1

例1: i=12;
a=++i; /* i=13 , a=13*/

例2: i=12;
a=i++; /* i=13 , a=12*/

例3: i=12;
a=--i; /* i=11 , a=11*/

例4: i=12;
a=i--; /* i=11 , a=12*/





§ 11 增、减1算符

- ++ 用作**前缀**,表示**先增1**再引用; 用作**后缀**,表示**先引用**再增1
- -- 用作**前缀**,表示**先减1**再引用; 用作**后缀**,表示**先引用**再减1

例5: a=1;
b=2;
r1=(a++)+(a++)*b; /*r1=1+1*2;*/
printf("%d,%d",r1,a); /*3,3*/

例6: a=1;
b=2;
r1=(++a)+(++a)*b; /*r1=3+3*2;*/
printf("%d,%d",r1,a); /*9,3*/



§ 11 增、减1算符

- ++ 用作**前缀**,表示**先增1**再引用; 用作**后缀**,表示**先引用**再增1
- -- 用作**前缀**,表示**先减1**再引用; 用作**后缀**,表示**先引用**再减1
- ++或--运算只能用于变量, 不适用于表达式
- 运算符的组合规则: 总是尽可能多地从左至右组成运算符

例7: `i=3;`
`k=(i++)+(i++)+(i++);` /*标准C平台: **i=6,k=9****/
`i=3;`
`k=(++i)+(++i)+(++i);` /*标准C平台: **i=6,k=18****/

例8: `i=3;`
`k=(++i)+(--i)+(++i);` /***i=4,k=12****/
`i=3;`
`k=(i++)+(++i)+i;` /***i=5,k=12****/

例9: `i=3;`
`printf("%d",-i++);` /***输出-3, i变为4****/



§ 11 增、减1算符

- ++ 用作**前缀**,表示**先增1**再引用; 用作**后缀**,表示**先引用**再增1
- -- 用作**前缀**,表示**先减1**再引用; 用作**后缀**,表示**先引用**再减1
- ++或--运算只能用于变量, 不适用于表达式
- 运算符的组合规则: 总是尽可能多地从左至右组成运算符

例10: 写出下列程序的运行结果

```
main( )  
{ int a,b;  
  a=b=5;  
  a=b+++7; /* a=12,b=6 */  
  a+=a-=a*a;  
  printf("a=%d,b=%d\n",a,b);  
}
```

a=-264,b=6



§ 12 逗号运算符,

1. 逗号表达式

表达式1, 表达式2

2. 从左至右计算, 结果类型和值是表达式2的类型和值

3. 逗号表达式可扩展成计算n个表达式, 逗号也称作顺序求值算符

表达式1, 表达式2,表达式n

eg1: $3+5, 6*8/-2$

eg3: `printf("%d\n", (t=3, t+2));`

eg2: `t=3, 6.0+t`

eg4: `x=(a=4, 6*2)` /*x=12, a=4, 结果=12*/

eg5: `x=a=4, 6*2` /*x=4, a=4, 结果=12*/

eg6: 假设所有变量均为整型, 则表达式

`(a=2, b=5, a++, b++, a+b)`的值为 9。





§ 12 逗号运算符,

例1：写出下列程序的运行结果

```
main( )
```

```
{ int a,x1,x2,x3;
```

```
  x1=(a=3,6*3); /* x1=18,a=3 */
```

```
  x2=a,6*3; /* x2=3 */
```

```
  printf(“%d,%d,%d”,x1,x2,x3=((a=3*5,a*4), a+5)); }
```

18,3,20

例2： $b=((b=(2,3),b+2),15+b)$ 运算后， b 的值是 18 。

例3： $b+=b=((b=(2,3),b+2),15+b)$ 运算后， b 的值是 36 。



习题

题1：当需要定义一个函数时，以下标识符中_____不能用作函数名。

- A. scanf B. for C. _sum D. F2

题2：已有声明语句“unsigned char ch;”，与表达式“ch='B'”的值不相同的表达式是_____。(字符A的ASCII码以十进制表示为65)。

- A. ch=0x42 B. ch=0102 C. ch='\102' D. ch=066

题3：下列语句中，合法的语句是_____。

- A. a=1,b=2 B. ++a;
C. a=a+1=5; D. y=int(a);



习题

题4: 设x、y均为float型变量, 则以下不合法的赋值语句是_____。

A. ++x;

B. $y=(x\%2)/10$;

C. $x*=y+8$;

D. $x=y=0$;

题5: 已知int i; float x,y=3.7; 执行语句x=i=y; 后, 变量x的值是_____。

题6: 已知: char a; int b; float c; double d; 执行 $c=a+b+c+d$ 后, 变量c的类型是_____。

A. int

B. char

C. float

D. double

题7: 已知int x; 则使用逗号表达式 $(x=4*5, x+5)$, $x+25$ 后的结果是_____, x之值是_____。



习题

题8：设以下变量均为int型，则值不等于7的表达式是_____。

A. $(x=y=6, x+y, x+1)$

B. $(x=y=6, x+y, y+1)$

C. $(x=6, x+1, y=6, x+y)$

D. $(y=6, y+1, x=y, x+1)$

题9：已知int i=5;执行 $i+=++i$ 后，i之值是_____。

题10：int a=2,b=3,c=2; $a+=b*=(++b-c++)$;变量a的值是_____,
b的值是_____。

题11：假设m是一个三位数，从左到右用a、b、c表示各位的数字，则从左到右各个数字是bac的三位数的表达式是 $m/10\%10*100+m/100*10+m\%10$ 。



习题

题12：说出下列程序的功能和运行结果。

```
#define INCMCL j*=++i  
main( )  
{ int i=1,j=1;  
  INCMCL; /*j*=++i   j=1*2 */  
  INCMCL; /*j*=++i   j=1*2*3 */  
  INCMCL; /*j*=++i   j=1*2*3*4 */  
  INCMCL; /*j*=++i   j=1*2*3*4*5 */  
  printf("j=%d\n",j);  
}
```

功能：求5! 结果：j=120

