



中國人民大學
RENMIN UNIVERSITY OF CHINA

C程序设计

第3讲 变量运算与输入输出

余力

buaayuli@ruc.edu.cn

内容提要

3.1 基本数据类型

3.2 输入和输出

3.3 运算符与表达式

3.4 数据表示与类型转换



中國人民大學
RENMIN UNIVERSITY OF CHINA



1. 基本数据类型

数据类型

- C语言支持的数据类型
 - 整型数据 (int)
 - 浮点型数据 (float)
 - 字符型数据 (char)
 - 用户自定义数据 (pointer, struct, union)
- 为什么C语言要区分不同数据类型？

更细分的数据类型

	signed	unsigned
short	short int x; short y;	unsigned short x; unsigned short int y;
default	int x;	unsigned int x;
long	long x;	unsigned long x;
float	float x;	N/A
double	double x;	N/A
char	char x; signed char x;	unsigned char x;

- 有符号(signed)与无符号(unsigned)的区分
- 表示整数使用int, short, long三种类型
- 表示实数使用float, double两种类型
- 表示字符使用char类型

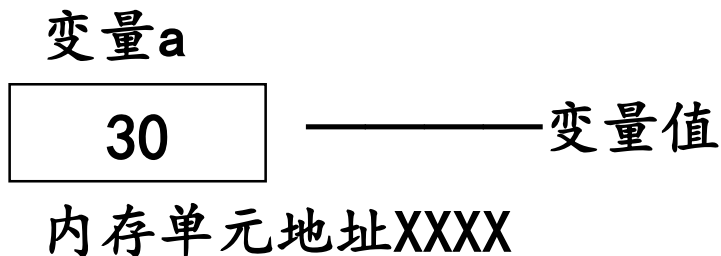
字符型数据

- 占用1个字节字符类型的数据
- 'a'的ASCII码为97，内存中存储形式：

0 1 1 0 0 0 0 1

ASCII 值	字符	控制字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
000	null	NUL	032	(space)	064	⑆	096	Ç	128	À	160	à	192	Ĺ	224	α
001	☉	SOH	033	!	065	À	097	à	129	Á	161	á	193	Ľ	225	β
002	☼	STX	034	"	066	B	098	b	130	Â	162	â	194	Ť	226	Γ
003	▼	ETX	035	#	067	C	099	c	131	Ã	163	ã	195	Ŧ	227	δ
004	✦	EOT	036	\$	068	D	100	d	132	Ä	164	ä	196	—	228	Σ
005	♣	END	037	%	069	E	101	e	133	Å	165	å	197	†	229	σ
006	♠	ACK	038	&	070	F	102	f	134	Æ	166	æ	198	‡	230	μ
007	beep	BEL	039	'	071	G	103	g	135	Ç	167	ç	199	‡	231	τ
008	backspace	BS	040	(072	H	104	h	136	È	168	è	200	¼	232	ϕ
009	tab	HT	041)	073	I	105	i	137	É	169	é	201	½	233	θ
010	换行	LF	042	*	074	J	106	j	138	Ê	170	ê	202	¾	234	Ω
011	♂	VT	043	+	075	K	107	k	139	Ë	171	ë	203	⁄	235	δ
012	♀	FF	044	,	076	L	108	l	140	Ì	172	ì	204	℥	236	∞
013	回車	CR	045	-	077	M	109	m	141	Í	173	í	205	=	237	⊗
014	♪	SO	046	.	078	N	110	n	142	Î	174	î	206	≠	238	€
015	☼	SI	047	/	079	O	111	o	143	Ï	175	ï	207	≡	239	∩
016	▶	DLE	048	0	080	P	112	p	144	Ê	176	ë	208	≡	240	≡
017	◀	DC1	049	1	081	Q	113	q	145	Ë	177	ë	209	≡	241	≡
018	⋮	DC2	050	2	082	R	114	r	146	Ë	178	ë	210	≡	242	≡
019	!!!	DC3	051	3	083	S	115	s	147	Ë	179	ë	211	≡	243	≡
020	↑	DC4	052	4	084	T	116	t	148	Ë	180	ë	212	≡	244	≡
021	§	NAK	053	5	085	U	117	u	149	Ë	181	ë	213	≡	245	≡
022	■	SYN	054	6	086	V	118	v	150	Ë	182	ë	214	≡	246	≡
023	⋮	ETB	055	7	087	W	119	w	151	Ë	183	ë	215	≡	247	≡
024	↑	CAN	056	8	088	X	120	x	152	Ë	184	ë	216	≡	248	≡
025	↓	EM	057	9	089	Y	121	y	153	Ë	185	ë	217	≡	249	≡
026	→	SUB	058	:	090	Z	122	z	154	Ë	186	ë	218	≡	250	≡
027	←	ESC	059	;	091	[123	{	155	Ë	187	ë	219	≡	251	≡
028	⌂	FS	060	<	092	\	124		156	Ë	188	ë	220	≡	252	≡
029	•	GS	061	=	093]	125	}	157	Ë	189	ë	221	≡	253	≡
030	▲	RS	062	>	094	^	126	~	158	Ë	190	ë	222	≡	254	≡
031	▼	US	063	?	095	_	127	¸	159	Ë	191	ë	223	≡	255	Blank'FF'

变量



- 常量：程序执行过程中不发生变化
- 变量：在程序中可以被赋值而发生变化
- 变量的命名规则
 - 包含字母、数字和 '_'
 - 只能以字母或下划线开头
 - 关键字不能作为变量名 (for, if, while)
 - 大小写敏感
 - 变量在一个函数范围内不能重名

思考：变量命名

- 思考：指出下面变量命名的正确性
 - `int money$owed;`
 - `int total_count;`
 - `int score2;`
 - `int 2ndscore;`
 - `int long;`
 - `int x, X;` 定义了几个变量？

常量

- 整型、浮点型、字符型、字符串
- 整型常量
 - 八进制：以0开头的数字序列（注意是数字 0，不是字母 o）
 - 十六进制表示：以0x开头的数字序列
 - 十进制：其它数字序列。

	short	int	long
八进制	%ho	%o	%lo
十进制	%hd	%d	%ld
十六进制	%hx 或者 %hX	%x 或者 %X	%lx 或者 %lX

常量的书写格式

- 长整型数据要在数字后加字母L或l。例:

-12L (十进制)

774545L (十进制)

076L (八进制 , 62)

0100000L (八进制 , 32768)

0X12l (十六进制 , 18)

0x8000l (十六进制 , 32768)

```
int main() {  
    int a, b;  
    a = 015;  
    printf("%3d\n", a);  
    printf("%3o\n", a);  
    printf("%3x\n", a);  
    return 0;  
}
```

+Chp03_进制.cpp

- 注意：12L与12值相等，但占用的存储空间不同。

实型常量

- 实型常量可以用小数或指数表示，例：

34.5 3.14 .345 345. 1e2 1.5e-3

- 字母e(或E)之前**必须有数字**，且e后指数必须为**整数**
- 小数点左边应有一位（且只能有一位）非零的数字.
- 例：123.456可以表示为：123.456**e0**， 12.3456**e1**，
0.123456**e3**， 0.0123456**e4**

```
int main() {  
    float a, b;  
    a = 123.456e2;  
    printf("%d\n", (int)a);  
    printf("%f\n", a);  
    return 0;  
}
```

字符常量

- 字符型常量是用单撇号括起来的一个字符,如:

- 'a' 'A' '?' '#' '8' " ' '

- 转义字符

字符形式	功能
\n	换行
\t	横向跳格
\v	竖向跳格
\b	退格
\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 ASCII 码
\xhh	1 到 2 位 16 进制 ASCII 码

字符串

- 字符串：

I say:'Goodby!'

可写成：

"I say:\'Goodby!\'"

- 字符串"hello"存储为：

h	e	l	l	o	\0
---	---	---	---	---	----

- 实际上每个字符都以ASCII码存储

104	101	108	108	111	0
-----	-----	-----	-----	-----	---

符号常量的使用

//已知单价和数量，求总价

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

```
#define PRICE 30
```

```
int main( )
```

```
{
```

```
    int num, total;
```

```
    num = 10;
```

```
    total = num * PRICE;
```

```
    printf( "total = %d\n", total);
```

```
    return 0;
```

```
}
```

- #define是宏定义命令
- 在本程序中，用PRICE来替代30这个数字
- 宏定义不是语句，结尾没有;



中國人民大學
RENMIN UNIVERSITY OF CHINA



2. 格式输入输出

设计 printf 的基本思想

- 需要考虑哪些基本问题
 - What - 要输出的是什么？
 - How - 要怎样进行输出？

格式控制

变量列表



- “You look great in %s\n”

输出函数 printf 的使用

■ “格式声明” 的使用

➤ 格式声明 = % + 附加字符 + 格式字符

对数据输出
做基本的“排版”

不同类型变量
输出格式与精度

➤ 常用格式字符

- 输出整数：`%4d`; `%u`, `%o`, `%x`, `%X`

- 输出实数：`%f`, `%-n.mf`, `%e`, `%E`

- 输出单个字符：`%c`

- 输出字符串：`%s`

```
long a;  
scanf("%ld",&a);  
printf("%ld",a);
```

```
double f;  
scanf("%lf", &f );  
printf("%lf", f );
```

输出数据的最小宽度：一个整数`n`

靠左对齐与靠右对齐：是否加符号-

思考

- 声明整型变量age，将变量赋值为你年龄，使用三次printf()函数
 - 使用一个printf()函数，将你的姓名和age，打印在一行
 - 使用一个printf()函数，将你的姓名和age，打印在两行
 - 使用两个printf()函数，将你的姓名和age，打印在一行

+ Chp03_printf分行输出.cpp

四舍五入

```
int main() {  
    float arr[3] = { 3.5, 3.4, 3.7 };  
    printf("将arr内容四舍五入后:\n");  
    for (int i = 0; i < 3; i++) {  
        printf("%d\n", (int)(arr[i] + 0.5));  
    }  
}
```

```
int main() {  
    float num = 3.1415;  
    num = (int)(num * 1000 + 0.5) / 1000.0;  
    printf("%.3f", num);  
}
```

```
int main() {  
    float num = 3.14159;  
    num = round(num * 10000) / 10000.0;  
    printf("%.4f", num);  
}
```

+Chp03_四舍五入.cpp

设计 scanf 的基本思想

- 需要考虑哪些基本问题
 - What - 要输入的是什么？
 - How - 要怎样进行输入？
 - 参考 printf 的例子，scanf 也应包含
 - 格式控制，如"%d %f"
 - 变量列表，如&num, &score
 - 字符串怎么输入？
 - 示例：scanf ("%s", name)
- 字符串不能包含空格**

getchar与gets

- gets读一行，getchar()读一个； 输出：putchar(ch1); puts()

```
int main() {  
    char ch1, ch2;  
    ch1 = getchar();  
    getchar();  
    ch2 = getchar();  
    printf("%c %c\n", ch1, ch2);  
    return 0;  
}
```

```
int main() {  
    char str1[20], str2[20];  
    scanf("%s", str1);  
    printf("%s\n", str1);  
    gets(str2);  
    printf("%s\n", str2);  
  
    return 0;  
}
```

可以输入空格

+Chp03_getchar_gets.cpp

C++的输入输出流

```
#include<iostream>
using namespace std;
```

```
int main () {
    int n1,n2;
    float f1, f2;
    double d1, d2;
    char c1, c2;

    cin >> n1 >> n2;
    cin >> f1 >> f2;
    cin >> d1 >> d2;
    cin >> c1 >> c2;

    cout << n1 << " " << n2 << endl;
    cout << f1 << " " << f2 << endl;
    cout << d1 << " " << d2 << endl;
    cout << c1 << " " << c2 << endl;
    return 0;
}
```

What is new?

- 建立C++项目与.cpp为扩展名的源文件
 - C++兼容C语言（语句、头文件.....）
- 预处理：引入输入输出流的头文件

```
#include<iostream>  
using namespace std;
```

- 输入流cin
 - 从键盘提取变量信息（>>）
- 输出流cout
 - 将变量插入到显示设备中（<<）

输出流的基本操作

- cout语句的一般格式为

- cout<<表达式1<<表达式2<<.....<<表达式n;

- 一个cout语句可以分成若干行写

cout << n1 << " " << n2 << endl << f1 << " " << f2 << endl;



等价

cout << n1 << " " << n2 << endl;

cout << f1 << " " << f2 << endl;

```
1  cout<<"This is " //注意行末尾无分号
2  <<"a C++ "
3  <<"program."
4  <<endl; //语句最后有分号
```

等价

```
1  cout<<"This is "; //语句末尾有分号
2  cout <<"a C++ ";
3  cout <<"program.";
4  cout<<endl;
```


输入流的基本格式

- cin语句的一般格式为

- cin>>变量1>>变量2>>.....>>变量n;

- 一个cin语句可以分成若干行写

```
cin >> n1 >> n2 >> f1 >> f2  
>> d1 >> d2 >> c1 >> c2;
```

等价



```
cin >> n1 >> n2;  
cin >> f1 >> f2;  
cin >> d1 >> d2;  
cin >> c1 >> c2;
```

- 在用cin输入时,系统也会根据变量的类型从输入流中提取相应长度的字节

- **不能**用cin语句把**空格字符**和**回车换行符**作为字符输给**字符变量**, 它们被**跳过**

用输入流接收字符串

```
1  #include <iostream>
2  using namespace std;
3  main ()
4  {
5      char a[20]; //此处也可以是char* a;
6      cin>>a;
7      cout<<a<<endl;
8  }
```

- 接收字符串时，遇“空格”、“TAB”、“回车”都结束。

如何接收一行字符串

- 一行字符串可能包含空白

```
1 # include <iostream>
2 using namespace std;
3
4 □ int main() {
5     char m[20];
6     cin.getline(m, 3);
7     cout << m << endl;
8 }
```

- **cin.getline** (接受字符串的数组m, 接受个数3, 结束字符)
 - 当第三个参数省略时, 系统默认为 ' \0 '

接收单个字符

```
1 # include <iostream>
2 using namespace std;
3
4 int main() {
5     char ch;
6     cin.get(ch);
7     // ch= cin.get();
8     cout << ch << endl;
9 }
```

- **cin.get(字符变量名)**
 - 可以用来接收字符，可以接收空格、TAB和回车。



中國人民大學
RENMIN UNIVERSITY OF CHINA



3. 运算符与表达式

算术运算符

- 基本的算术运算符

- 正(负)号：+ (-)

- 加： +

- 减： -

- 乘： *

- 除： /

- 求余： %

- 例子：如求21整除4的余数，可用下式表示：21%4

自增、自减运算符

- $++i, --i$
 - 在使用之前，先使 i 的值加(减)1
- $i++, i--$
 - 使用之后，使 i 的值加(减)1
- 要先彻底理解，再进行使用

运算符

- 关系运算符

- $>$, $>=$, $<$, $<=$, $==$, $!=$

- 逻辑运算符

- $!$, $\&\&$, $||$

- 赋值运算符

- $=$

运算符的优先次序（参考）

- | | |
|--|---------------------------------------|
| 1. <code>., →, [], ()</code> | 8. <code>&</code> |
| 2. <code>++, --, !, ~,</code>
<code>(DataType), sizeof</code> | 9. <code>^</code> |
| 3. <code>*, /, %</code> | 10. <code> </code> |
| 4. <code>+, -</code> | 11. <code>&&</code> |
| 5. <code>>>, <<</code> | 12. <code> </code> |
| 6. <code>>, <, >=, <=</code> | 13. <code>?:</code> |
| 7. <code>==, !=</code> | 14. <code>=, +=, -=, *=,</code> |

不用死记，复杂表达式中多用 “()”运算符

算术表达式

- 例如：20+30，30-10，30*45，45/15, 45/10
- 整数与整数运算结果为整数
- 整数与小数？30+12.3
- 平方：12*12，
- 开方：100的平方根
- 指数次方：
- 余弦、正弦、绝对值、...

函数名	函数原型	功 能
abs	int abs (int x);	求整数 x 的绝对值
acos	double acos (double x);	计算 $\cos^{-1}(x)$ 的值
asin	double asin (double x);	计算 $\sin^{-1}(x)$ 的值
atan	double atan (double x);	计算 $\tan^{-1}(x)$ 的值
atan2	double atan2 (double x, double y);	计算 $\tan^{-1}(x/y)$ 的值
cos	double cos(double x);	计算 $\cos(x)$ 的值
cosh	double cosh(double x);	计算 x 的双曲余弦 $\cosh(x)$ 的值
exp	double exp(double x);	求 e^x 的值
fabs	double fabs(double x);	求 x 的绝对值
floor	double floor(double x);	求出不大于 x 的最大整数
fmod	double fmod(double x, double y);	求整除 x/y 的余数
frexp	double frexp (double val, int * eptr);	把双精度数 val 分解为数字部分(尾数) x 和以 2 为底的指数 n , 即 $val = x * 2^n$, n 存放在 $eptr$ 指向的变量中

函数名	函数原型	功 能
log	double log(double x);	求 $\log_e x$, 即 $\ln x$
log10	double log10(double x);	求 $\log_{10} x$
modf	double modf (double val, double * iptr);	把双精度数 val 分解为整数部分和小数部分, 把整数部分存到 $iptr$ 指向的单元
pow	double pow (double x, double y);	计算 x^y 的值
rand	int rand(void);	产生 -90~32 767 的随机整数
sin	double sin(double x);	计算 $\sin x$ 的值
sinh	double sinh(double x);	计算 x 的双曲正弦函数 $\sinh(x)$ 的值
sqrt	double sqrt(double x);	计算 \sqrt{x}
tan	double tan(double x);	计算 $\tan(x)$ 的值
tanh	double tanh(double x);	计算 x 的双曲正切函数 $\tanh(x)$ 的值

关系表达式

- 用**关系运算符**将两个表达式连接起来的式子，称关系表达式
- 关系表达式的值是一个逻辑值，即“真”或“假”

运算符	说明	
>	大于	优先级相同（高）
>=	大于等于	
<	小于	
<=	小于等于	
= =	等于	优先级相同（低）
!=	不等于	

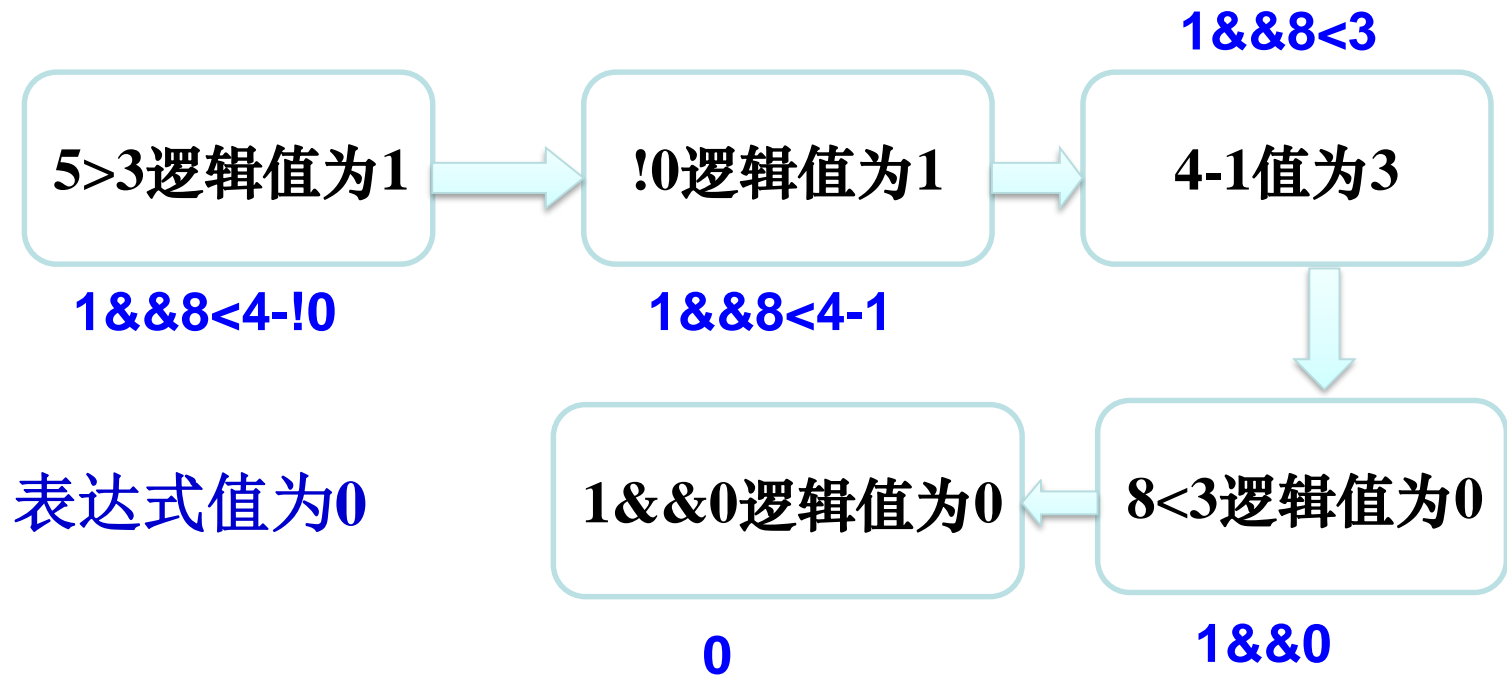
逻辑运算

运算符	说明
!	逻辑非 (NOT)
&&	逻辑与 (AND)
	逻辑或 (OR)

- $! > \&\& > ||$ ，即 “!”为三者中最高。
- “&&”和 “||”**低于**关系运算符。
- “!”**高于**算术运算符。

逻辑表达式 (1)

- 用逻辑运算符将关系表达式或逻辑量连接起来的式子
- 求解： $5>3\&\&8<4-!0$ 的值



逻辑表达式 (2)

■ 并不是所有逻辑运算符都要被执行

- **$a \& \& b \& \& c$** 只有a为真时，才需要判断b的值，只有a和b都为真时，才需要判断c的值。
- **$a || b || c$** 只要a为真，就不必判断b和c的值，只有a为假，才判断b。a和b都为假才判断c
- 例如： $(m = a > b) \& \& (n = c > d)$
 - $a=1, b=2, c=3, d=4$, m和n的原值为1时，由于“ $a > b$ ”的值为0，因此 $m=0$ ，而“ $n = c > d$ ”不被执行，因此n的值不是0而仍保持原值1。

赋值表达式

- $\langle \text{变量} \rangle \langle \text{赋值运算符} \rangle \langle \text{表达式} \rangle$
 - $N = 20;$
- 先求赋值表达式右侧表达式的值，然后赋值给运算符左侧的变量
- 赋值表达式的值是赋值号左边变量被赋值后的值
- **复合赋值运算符**
 - $N += 6;$



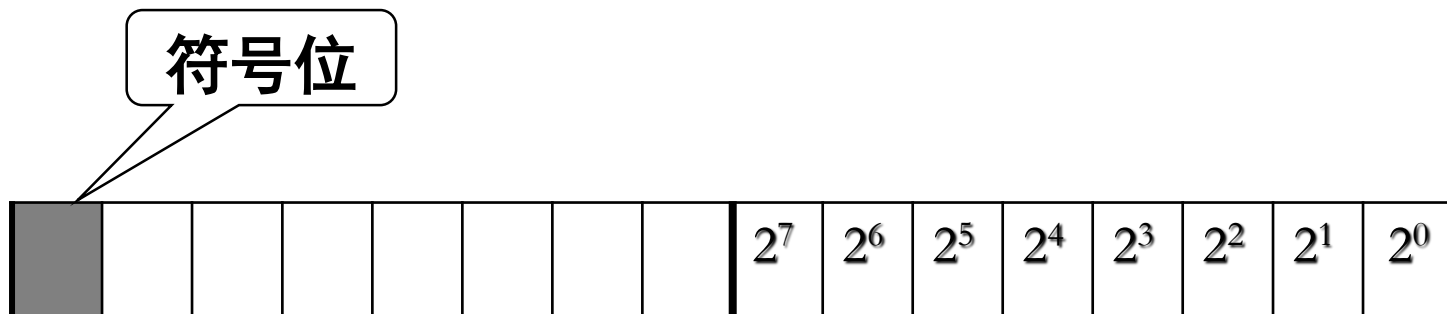
中國人民大學
RENMIN UNIVERSITY OF CHINA



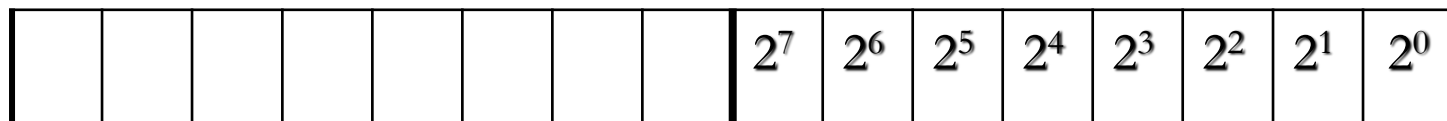
4. 数据存储与类型转换

整数类型 – 符号

16位有符号整数



16位无符号整数



整数类型 – 数值范围

类型	符号	字节	位数	最小值	最大值
short int	Signed	2	16	-32768	32767
	unsigned			0	65535
int	Signed	4	32	-2,147,483,648	2,147,483,647
	unsigned			0	4,294,967,295
long int	Signed	4	32	-2,147,483,648	2,147,483,647
	unsigned			0	4,294,967,295

整数类型 – 存储空间

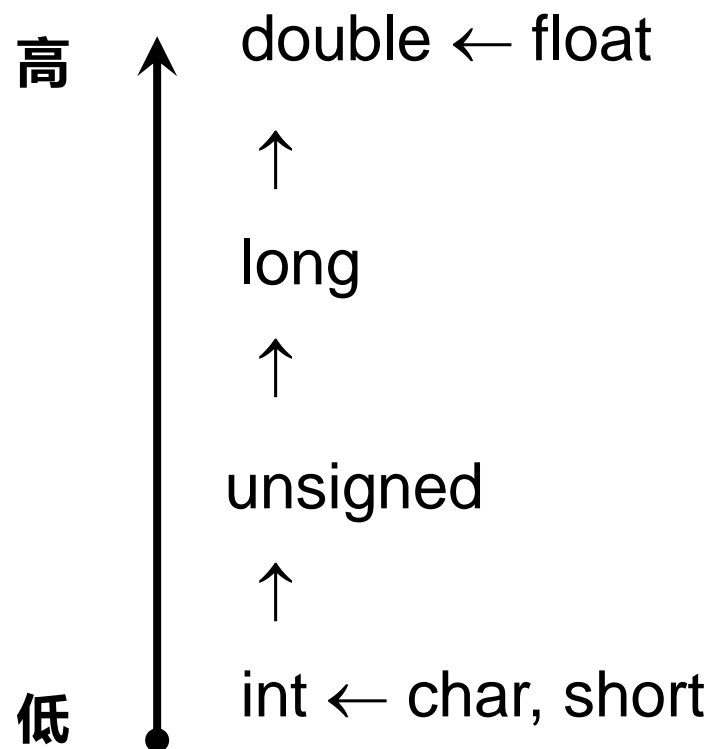
```
#include <stdio.h>

int main() {
    printf( "char: %d bytes \n", sizeof(char) );
    printf( "short: %d bytes \n", sizeof(short) );
    printf( "int: %d bytes \n", sizeof(int) );
    printf( "long: %d bytes \n", sizeof(long) );
    printf( "float: %d bytes \n", sizeof(float) );
    printf( "double: %d bytes \n", sizeof(double) );
    return 0;
}
```

+Chp03_sizeof.cpp

char: 1 bytes
short: 2 bytes
int: 4 bytes
long: 4 bytes
float: 4 bytes
double: 8 bytes

1、一般算术转换



类型的高低

2、赋值转换

- C语言允许通过赋值使 “=”右边表达式的值的类型自动转换为其左边变量的类型。
- 赋值转换具有强制性，可能是提升，也可能是降格。

3、输出转换

- 转换为与原有类型不同的类型输出。例如：
 - long型数据在printf函数中指定用int型输出
 - int型数可按无符号方式输出（使用%u转换）

■ 但不建议跨类型这样！！！！

■ 小数以整数格式printf输出有问题

```
int main() {  
    long a = 123L;  
    printf( "%d\n", a);  
  
    int b = 123;  
    printf( "%ld\n", b);  
  
    float x = 12.87;  
    printf( "%d\n", x);  
  
    int y = 12;  
    printf( "%f\n", y);  
  
    return 0;  
}
```

4. 强制转换

- C语言提供一种“强制类型转换”运算符，将一种类型的变量强制转换为另一种类型。
- 一般形式为：

(类型标识符) 表达式

例：

(char)(3 - 3.14159 * x)

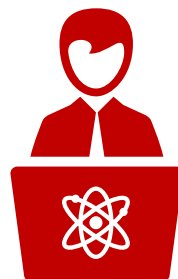
k=**(int)**((int)x + (float)i + j)

(float)(x = 99)

(double)(5 % 3)



中國人民大學
RENMIN UNIVERSITY OF CHINA



谢谢大家！

