

第3章 数据描述

一 数据类型运算符与体现式

- 预备知识
- 数据类型
- 常量与变量
- 不同类型数据间的转换
- 运算符和体现式



■ 3.0 预备知识

□ 计算机中数的表达及进制转换

■ 数码、基与权

- 数码：表达数的符号
- 基：数码的个数
- 权：每一位所具有的值

■ 数制



■ 多种进制之间的转换

□ 二进制、八进制、十六进制转换成十进制

■ 措施：按权相加

例 $(111011)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (59)_{10}$

例 $(136)_8 = 1 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 = (94)_{10}$

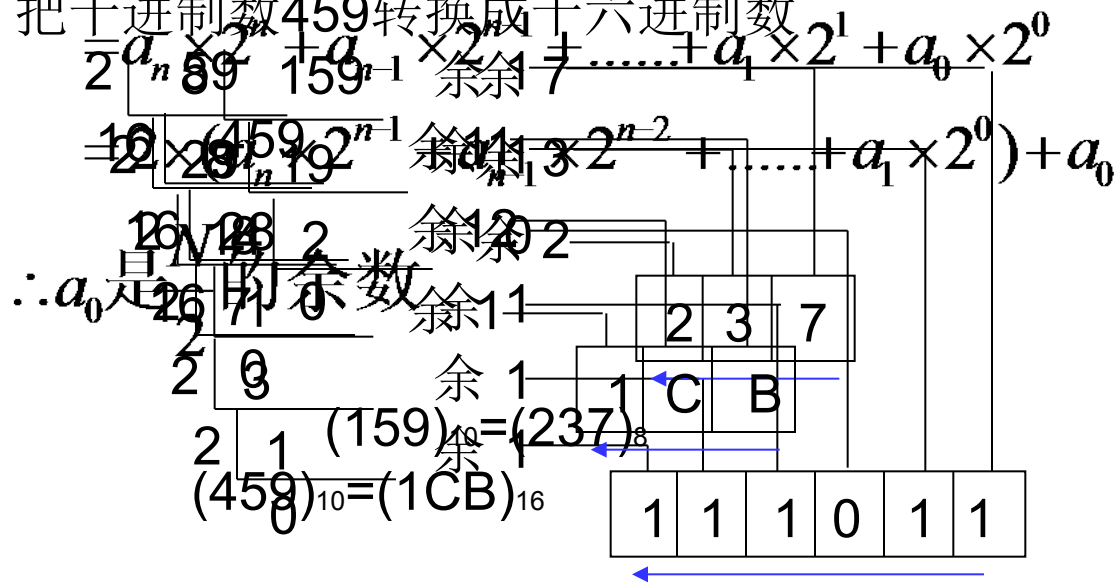
例 $(1F2A)_{16} = 1 \times 16^3 + 15 \times 16^2 + 2 \times 16^1 + 10 \times 16^0 = (7978)_{10}$



■ 多种进制之间的转换(整数)

- 二进制、八进制、十六进制转换成十进制
 - 措施：按权相加
- 十进制转换成二进制、八进制、十六进制
 - 原理：
 - 措施：连续除以基，从低到高统计余数，直至商为0

例 把十进制数 459 转换成十六进制数
 例 把十进制数 59 转换成二进制数



$(59)_{10} = (111011)_2$

□ 二进制与八进制之间的转换

- 二进制转换成八进制：从右向左，每3位一组（不足3位左补0），转换成八进制
- 八进制转换成二进制：用3位二进制数替代每一位八进制数

例 $(1101001)_2 = (001, 101, 001)_2 = (151)_8$

例 $(246)_8 = (010, 100, 110)_2 = (10100110)_2$

000	~	0
001	~	1
010	~	2
011	~	3
100	~	4
101	~	5
110	~	6
111	~	7



□ 二进制与十六进制之间的转换

- 二进制转换成十六进制：从右向左，每4位一组（不足4位左补0），转换成十六进制
- 十六进制转换成二进制：用4位二进制数替代每一位十六进制数

$$_{2}=(0011,0101,0111,1101)_{2}=(357D)_{16}$$

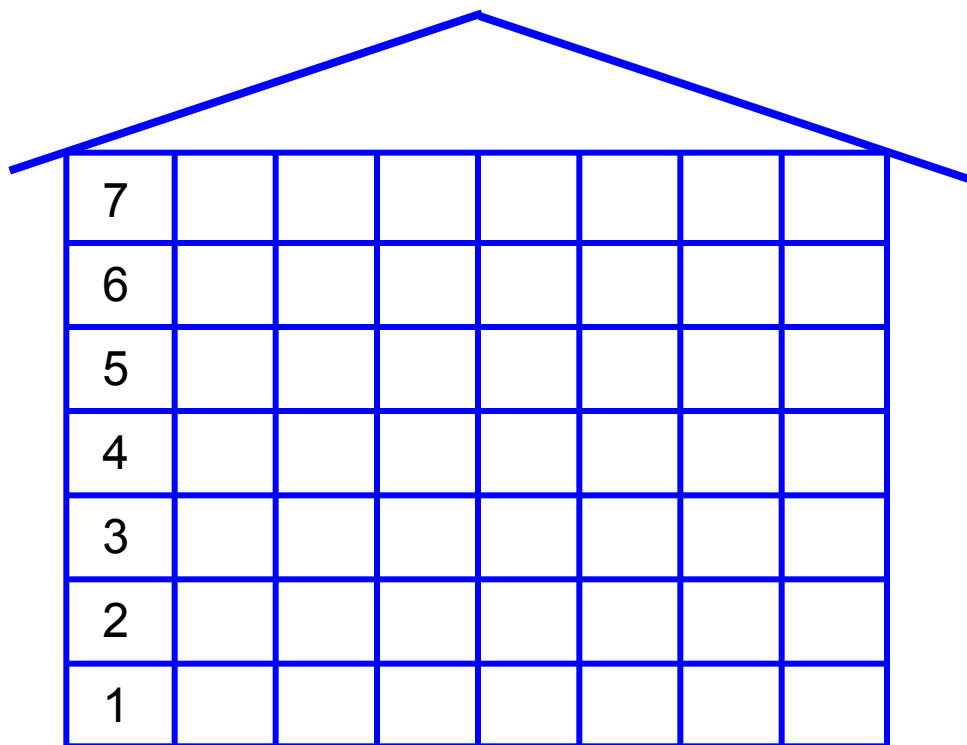
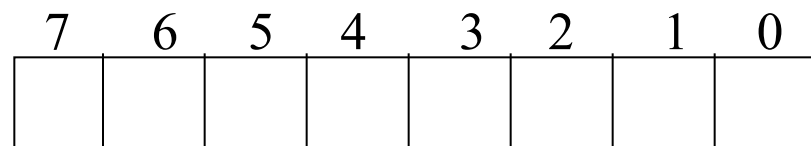
$$\text{例 } (4B9E)_{16}=(0100,1011,1001,1110)_{2}$$

0000	~	0
0001	~	1
0010	~	2
0011	~	3
0100	~	4
0101	~	5
0110	~	6
0111	~	7
1000	~	8
1001	~	9
1010	~	A
1011	~	B
1100	~	C
1101	~	D
1110	~	E
1111	~	F



□ 字节和位

- 内存以字节为单元构成
- 每个字节有一种地址
- 一种字节一般由8个二进制位构成
- 每个二进制位的值是0或1



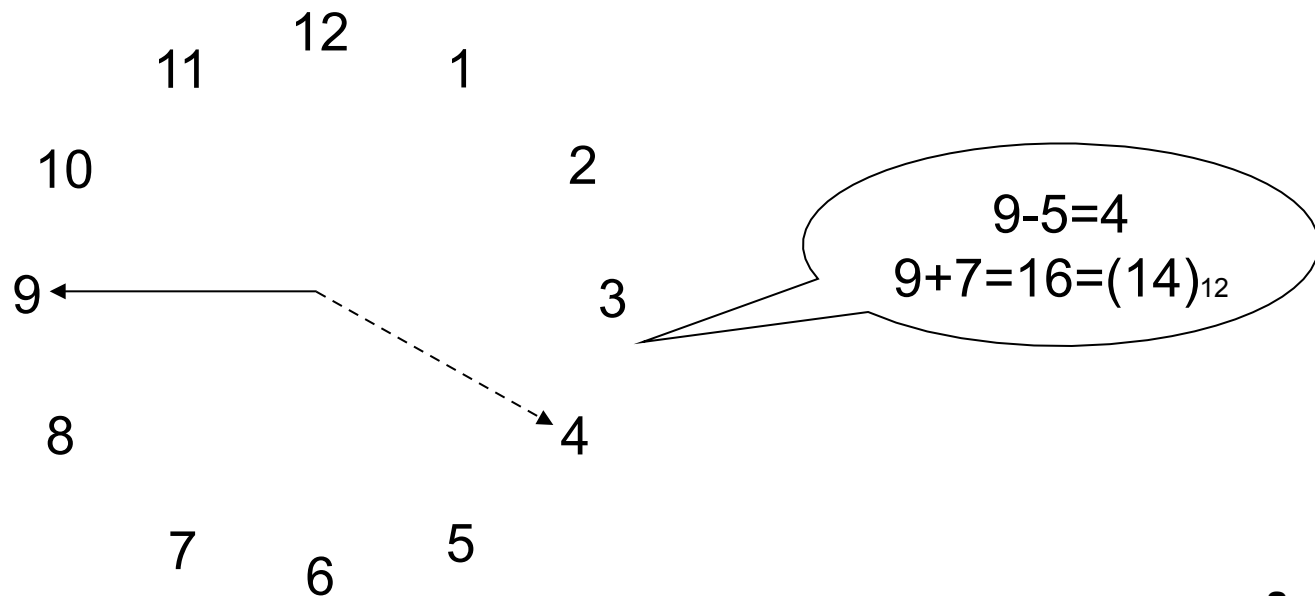
0
1
2
3
4
5
6
7
8
9
10

⋮



□ 数值的表达措施——原码、反码和补码

- 原码：最高位为符号位，其他各位为数值本身的绝对值
- 反码：
 - 正数：反码与原码相同
 - 负数：符号位为1，其他位对原码取反
- 补码：
 - 正数：原码、反码、补码相同
 - 负数：最高位为1，其他位为原码取反，再对整个数加1



(用一字节表达数)

	原码	反码	补码
+7	00000111	00000111	00000111
-7	10000111	11111000	11111001
+0	00000000	00000000	00000000
-0	10000000	11111111	00000000
数的范围	01111111~ 11111111 (-127~+127)	01111111~ 10000000 (-127~+127)	01111111~ 10000000 (-128~+127)

- 负数补码转换成十进制数：最高位不动，其他位取反加1

例 补码：11111001
取反：10000110
加1：10000111=-7



浮点表达

- 十进制数的科学表达法 (scientific notation)

$$156,000,000 \Rightarrow 1.56 * 10^8$$

$$-0.000,000,28 \Rightarrow 0.28 * 10^{-6}$$

- 浮点表达

$$N = M * R^E$$

M: 尾数, R: 基数, E: 指数 (阶码)

数的符号位在M中表达;

数的大小在E中表达。

补码

- 在给定码长 N 的情况下, 求数 $-n$ 的补码, 就是求一种二进制数 n' , 使得 $n + n' = 0 = 2^N$

- 对于整数 X 的补码 $[X]$ 码长, 则
$$[X] = \begin{cases} X & : 0 \leq X \leq 2^{N-1} - 1 \\ 2^N - |X| & : -2^{N-1} \leq X < 0 \end{cases}$$

- 补码只有一种0的表达, 即 $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$

按位求补码

- 正数补码: 和原码相同
- 负数补码: 反码加 1

例: 码长为8, 求-25的补码

原码	10011001
反码	11100110
加1	11100111

补码运算示例

■ 码长4位, 求 $2 - 3$

+2的补码 0010

-3的补码 1101

成果 1111

■ 码长4位, 求 $3 - 2$

+3的补码 0011

-2的补码 1110

成果 0001

3.1 数据类型

□ 数据类型总表

C 数据类型

基本类型

整型

短整型short

整型int

长整型long

实型

单精度型float

双精度型double

字符类型char

数组

构造体struct

共用体union

枚举类型enum

构造类型

指针类型

空类型void

定义类型typedef

数据类型决定:

1. 数据占内存字节数
2. 数据取值范围
3. 其上可进行的操作



□ 基本数据类型

类型	符号	关键字	所占位数	数的表达范围
整型	有	(signed)int	16	-32768~32767
		(signed)short	16	-32768~32767
		(signed)long	32	-2147483648~2147483647
	无	unsigned short	16	0~65535
		unsigned int	16	0~65535
		unsigned long	32	0~4294967295
实型	有	float	32	3.4e-38~3.4e38
	有	double	64	1.7e-308~1.7e308
字符型	有	char	8	-128~127
	无	unsigned char	8	0~255

阐明:数据类型所占字节数随机器硬件不同而不同,上表以IBM PC机为例:



3.2 常量与变量

□ 标识符

- 定义：用来标识变量、常量、函数等的字符序列
- 构成：
 - 只能由字母、数字、下划线构成，且第一种字母必须是字母或下划线
 - 大小写敏感
 - 不能使用关键字
- 长度：最长32个字符

例：判断下列标识符号正当性

sum Sum M.D.John day Date **3days**
student_name **#33** lotus_1_2_3
char a>b _above **\$123**



□ 常量

- 定义：程序运营时其值不能变化的量（即常数）

- 分类：

- 符号常量：用标识符代表常量

- 定义格式：`#define` 符号常量 常量

- 一般用大写字母

- 是如 `#define`

- 直接常量：

- 整型常量



- 实型常量



- 字符常量



- 字符串常量



例 符号常量举例(ch2_1.c)

```
#define PRICE 30
```

```
main()
```

```
{
```

```
    int num,total;
```

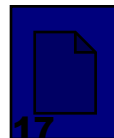
```
    num=10;
```

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

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

```
}
```

运营成果：total=300



📖 整型常量（整常数）

■ 三种形式：

- 十进制整数：由数字0~9和正负号表达. 如 123, -456, 0
- 八进制整数：由数字0开头, 后跟数字0~7表达. 如0123, 011
- 十六进制整数：由0x开头, 后跟0~9, a~f, A~F表达.
0x123, 0Xff

■ 整型常量的类型

- 根
- 在

问题：

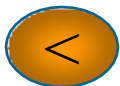
0123 = ()₁₀

0x123 = ()₁₀

0Xff = ()₁₀

数据类型

它是long int 型常量



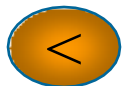
实型常量（实数或浮点数）

■ 表达形式：

- 十进制数形式：（必须有小数点） 如0.123, .123, 123.0, 0.0, 123.
- 指数形式：（e或E之前必须有数字；指数必须为整数）
如12.3e3 , 123E2, 1.23e4 , e^{-5} ,

■ 实型常量的类型

- 默认double型
- 在实型常量后加字母f或F，以为它是float 型



字符常量

- 定义:用单引号括起来的单个一般字符或转义字符.
- 字符常量如 'a' 'A' '?' '\n' '\101'
- 转义字符:反斜线后面跟一种字符或一种代码值表达

例 转义字符举例(ch2_001.c,ch2_004.c) —10

```
转 main()  
{  
    printf("\101 \x42 C\n");  
    printf("I say:\"How are you?\"\n");  
    printf("\C Program\\n");  
    printf("Turbo \'C\'");  
}
```

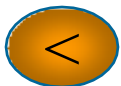
运营成果: (屏幕显示)
A B C
I say:"How are you?"
\C Program\
Turbo 'C'

反斜线

双引号

```
例 main()  
{ printf("Y\b=\n");  
}
```

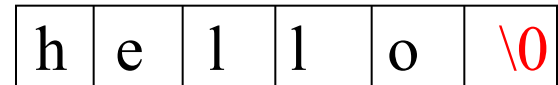
运营成果: 的字符
屏幕显示: =
打印机输出: ¥



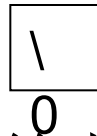
📖 字符串常量

- 定义：用双引号（“”）括起来的字符序列
- 存储：每个字符串尾自动加一种 ‘\0’ 作为字符串结束标志

例 字符串“hello”在内存中



例 空串 “”

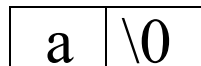


- 字符常量与字符串常量不同

例 ‘a’

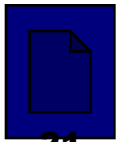
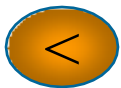


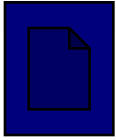
“a”



例：
char ch;
ch="A";

例：
char ch;
ch='A';





变量

- 概念：其值能够变化的量
- 变量名与变量值
- 变量定义的一般格式：

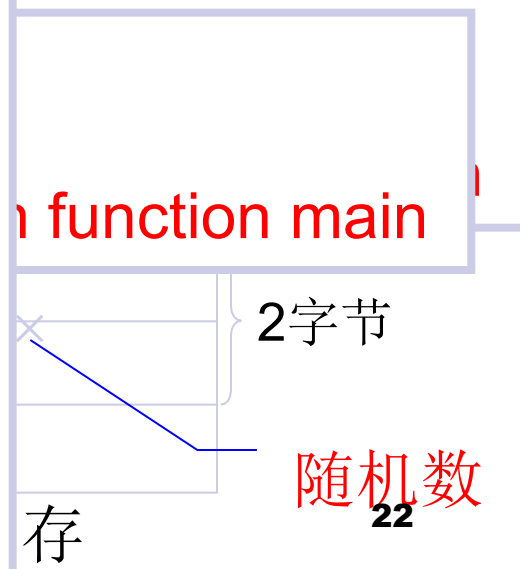
数据类型 变量1[, 变量2, ..., 变量n];

- 变量初始化: 定 例: `int a=1 b=-3 c;`
- 变量的使用: **先定义, 后使用** 例: `int a=2,b,c=4;`
- 变量的定义位置: **一般放在函数开头**

```

例2
float
c=a
main()
{
  { int a,b=2;
    a=1;
    float data;
    data=(a+b)*1.2;
    printf("data=%f\n",data);
  }
}

```



📖 整型变量

- 占字节数随机器不同而不同, 一般占一种机器字
- $\text{short} \leq \text{int} \leq \text{long}$
- 可用 `sizeof` (类型标识符) 测量

📖 实型变量

- `float`: 占4字节, 提供7位有效数字
- `double`: 占8字节, 提供15~16位有效数字

📖 字

```

例 float a;
a=111111.111; /* a=111111.1*/
double b;
b=111111.111; /* b=111111.111*/

```

没有字符串变量, 用字符数组存储

```

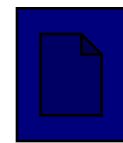
例 char a[5]; /* a="D"; */
char x[5]; /* x="A"; */

```

```

s = '!' + 'G' /* s=33+71; */

```



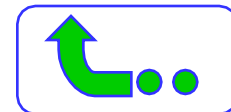
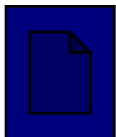
```
例 /*ch2_003.c*/  
#define PRICE 12.5  
main()  
{ int num=3;  
  float total;  
  char ch1,ch2='D';  
  
  total=num*PRICE;  
  ch1=ch2-'A'+ 'a';  
  printf("total=%f,ch1=%c\n",total,ch1);  
}
```

← 宏定义

} 变量定义

← 输出成果

运营成果：
total=37.500000, ch1=d



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/457101123154006156>