



北京大学
PEKING UNIVERSITY

信息科学技术学院

程序设计与算法(一)

C语言程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕!

讲义照片均为郭炜拍摄



北京大学
PEKING UNIVERSITY

信息科学技术学院

指定教材：

《新标准C++程序设计教程》

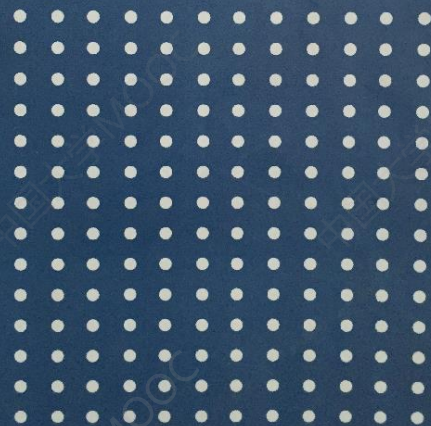
郭炜 编著

清华大学出版社

重点大学计算机专业系列教材

新标准C++程序设计教程

郭炜 编著



清华大学出版社



北京大学
PEKING UNIVERSITY

信息科学技术学院

二进制和十六进制



福建省宁德市北岸公园

用0和1表示各种信息

- 计算机的电路由逻辑门电路组成。一个逻辑门电路可以看成是一个开关，每个开关的状态是“开”（高电位）或“关”（低电位），即对应于1或0

用0和1表示各种信息

- 计算机的电路由逻辑门电路组成。一个逻辑门电路可以看成是一个开关，每个开关的状态是“开”（高电位）或“关”（低电位），即对应于1或0
- 二进制数的一位，取值只能是0或1，称为一个“比特” (bit)，简写：b

用0和1表示各种信息

- 计算机的电路由逻辑门电路组成。一个逻辑门电路可以看成是一个开关，每个开关的状态是“开”（高电位）或“关”（低电位），即对应于1或0
- 二进制数的一位，取值只能是0或1，称为一个“比特” (bit)，简写：b
- 八个二进制位称为一个“字节” (byte),简写: B

用0和1表示各种信息

- 计算机的电路由逻辑门电路组成。一个逻辑门电路可以看成是一个开关，每个开关的状态是“开”（高电位）或“关”（低电位），即对应于1或0
- 二进制数的一位，取值只能是0或1，称为一个“比特” (bit)，简写：b
- 八个二进制位称为一个“字节” (byte),简写: B
- $1024(2^{10})$ 字节称为1KB，1024KB称作1MB(1兆)，1024MB称作1GB，1024GB称作1TB。

用0和1表示各种信息

- 0和1足以表示和传播各种信息。

比如, 用8个连续的0或1 (即1个字节) 来表示一个字母、数字或标点符号, 比如用“00100000”表示空格, 用“01100001”表示字母“a”, 用“01100010”表示字母“b”, 用“01100011”表示字母“c”。由8个0或者1的组成的串, 一共有 2^8 即256种不同的组合, 这就足以表示10个阿拉伯数字以及英语中用到的所有字母和标点符号了。此即为ASCII编码方案。

- 图片、视频和可执行程序, 也可以用0和1表示

给定一个K进制数，求它是多大

假设有一个 $n+1$ 位的K进制数，它的形式如下：

$A_n A_{n-1} A_{n-2} \dots A_2 A_1 A_0$ (比如 八进制数 1723)

则其大小为：

$$A_0 \times K^0 + A_1 \times K^1 + \dots + A_{n-1} \times K^{n-1} + A_n \times K^n$$

数就是数，没有进制之分，只有数的表示形式，才有进制之分。
所谓“十进制数”，是“数的十进制表示形式”的简称。

给定一个数，求其K进制表示形式

● 求数的K进制表示形式 -- 短除法

给定一个整数N和进制K，那么N可表示成以下形式：

$$\begin{aligned} N &= A_0 \times K^0 + A_1 \times K^1 + A_2 \times K^2 + \dots + A_{n-1} \times K^{n-1} + A_n \times K^n \\ &= A_0 + K (A_1 + A_2 \times K^1 + \dots + A_{n-1} \times K^{n-2} + A_n \times K^{n-1}) \end{aligned}$$

N除以K所得到的余数是 A_0 ，商是 $A_1 + A_2 \times K^1 + \dots + A_{n-1} \times K^{n-2} + A_n \times K^{n-1}$ 。将这个商再除以K，就得到余数 A_1 ，新的商是

$$A_2 + A_3 \times K^1 + \dots + A_{n-1} \times K^{n-3} + A_n \times K^{n-2}$$

不停地将新得到的商除以K，直到商变成0，就能依次求得 A_0 、 A_1 、 A_2 …… A_{n-1} 、 A_n 。显然， $A_i < K$ ($i = 0 \dots n$)，且最终得到的K进制数就是：

$$A_n A_{n-1} A_{n-2} \dots A_2 A_1 A_0$$

K进制小数

K进制小数 $0.A_0A_1\dots A_n$ 的值是：

$$A_0 \times K^{-1} + A_1 \times K^{-2} + \dots + A_n \times K^{-(n+1)}$$

$$(0.12)_{10} = 1 \times 10^{-1} + 2 \times 10^{-2}$$

$$(0.1)_3 = 1 \times 3^{-1} \quad \text{即 } 1/3, \text{ 表示成10进制就是无限循环小数}$$

可见， n 进制下的有限位小数，在 m 进制下可能就无法精确表示，因为会无限循环

十进制有限位小数，在二进制的情况未必能用有限位数表示出来。计算机内存有限，不可能存放无限位，因此计算机的小数运算会有误差。比如，计算机其实无法精确表示 4.9，只能精确表示 4.899999999... 之类一个很接近的数

十六进制数

- 十六进制数应该有16个数字，除0到9外：

A 10

B 11

C 12

D 13

E 14

F 15

小写也可以

十六进制数到十进制数的转换

十六进制数	转换计算过程	十进制数
0	0×16^0	0
1	1×16^0	1
A	10×16^0	10
10	$0 \times 16^0 + 1 \times 16^1$	16
100	$0 \times 16^0 + 0 \times 16^1 + 1 \times 16^2$	256
AFD2	$2 \times 16^0 + 13 \times 16^1 + 15 \times 16^2 + 10 \times 16^3$	45010

十六进制数到二进制数的互相转换

4个二进制位正好对应于1个十六进制位

十六进制数	二进制数	十六进制数	二进制数
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

3AD → 11 1010 1101

110 1111 0010 → 6F2

101 → 5



北京大学
PEKING UNIVERSITY

信息科学技术学院

C++快速入门



美国科罗拉多大峡谷

为什么是C++而不是C语言

为什么是C++而不是C语言

- C语言是好东西，但是有点弱

为什么是C++而不是C语言

- C语言是好东西，但是有点弱
- C++ 更是好东西，但是有点烦

为什么是C++而不是C语言

- C语言是好东西，但是有点弱
- C++ 更是好东西，但是有点烦
- 我们要学的，是C++的一部分，基本上就是：
C语言+ STL（STL是C++中能让你节省大量编程时间的神兵！）
因为暂时不写大程序，因此不用关心“面向对象”的事情！

第一个C++程序

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    printf("Hello,world!");
    return 0;
}
```

输出:
Hello,world!

第一个C++程序

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    printf("Hello,world!");
    return 0;
}
```

蓝色部分暂时不用理会，照抄，
在 { 和 return 0; 之间写代码

Hello,world!

第一个C++程序

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    printf("Hello,world!");
    return 0;
}
```

蓝色部分暂时不用理会，照抄，
在 { 和 return 0; 之间写代码

“printf” 是用来执行输出的 “函数”

Hello,world!

第一个C++程序

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    printf("Hello,world!");
    return 0;
}
```

蓝色部分暂时不用理会，照抄，
在 { 和 return 0; 之间写代码

“printf” 是用来执行输出的 “函数”

Hello,world!

要输出的内容是一串文字（也叫字符串）
"Hello,world", C++规定，字符串前
后要用" "括起来。

第二个C++程序：输出更多

```
#include <cstdio>
#include <iostream>
using namespace std;
int main()
{
    int a = 3;
    printf("I have %d dollars.\n",a);
    printf("I want to buy:\na book.");
    return 0;
}
```

```
I have 3 dollars.
I want to buy:
a book.
```


第二个C++程序：输出更多

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a = 3;
    printf("I have %d dollars.\n",a);
    printf("I want to buy:\na book.");
    return 0;
}
```

a是“变量”，变量是用来存放数据的。
int 表示变量a是用来放整数的。

```
I have 3 dollars.
I want to buy:
a book.
```

第二个C++程序：输出更多

```
#include <cstdio>
#include <iostream>
using namespace std;
int main()
{
    int a = 3;
    printf("I have %d dollars.\n", a);
    printf("I want to buy:\na book.");
    return 0;
}
```

%d 表示要在此处输出一个整数，该整数的值是变量a的值。

```
I have 3 dollars.
I want to buy:
a book.
```

第二个C++程序：输出更多

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a = 3;
    printf("I have %d dollars.\n",a);
    printf("I want to buy:\na book.");
    return 0;
}
```

- `\n` 表示换行，它导致后面的输出会从下一行开始。

```
I have 3 dollars.
I want to buy:
a book.
```

第二个C++程序：输出更多

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a = 3;
    printf("I have %d dollars.\n",a);
    printf("I want to buy:\na book.");
    return 0;
}
```

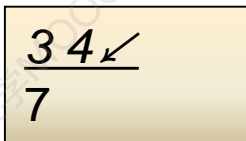
- 每条语句后面都要有 ;

```
I have 3 dollars.
I want to buy:
a book.
```

第三个C++程序：如何输入

输入两个整数，输出它们的和 (NOI.P0J 7883)

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a,b;
    scanf ("%d%d", &a, &b) ;
    printf ("%d", a+b) ;
    return 0;
}
```



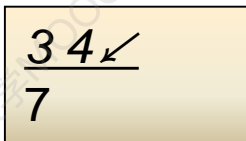
34 ✓
= 7

第三个C++程序：如何输入

输入两个整数，输出它们的和 (NOI.P0J 7883)

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a,b;
    scanf ("%d%d", &a, &b) ;
    printf ("%d", a+b) ;
    return 0;
}
```

a,b是“变量”，变量是用来存放数据的。
int 表示变量a,b是用来放整数的。



3 4 ✓
7

第三个C++程序：如何输入

输入两个整数，输出它们的和 (NOI.P0J 7883)

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a,b;
    scanf ("%d%d", &a, &b) ;
    printf ("%d", a+b) ;
    return 0;
}
```

a,b是“变量”，变量是用来存放数据的。
int 表示变量a,b是用来放整数的。

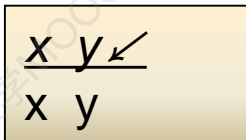
scanf是能执行输入功能的一个“函数”。
%d 表示要等待输入一个整数
&a 表示要把输入的整数放入变量a

3 4 ✓
7

输入字符

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    char a,b,c;
    scanf ("%c%c%c", &a, &b, &c) ;
    printf ("%c%c%c", a, b, c) ;
    return 0;
}
```

%c表示读入一个字符，不会跳过空格



程序的注释

- 经常需要在程序中写点说明性的文字，帮助程序阅读者理解程序。这就需要用注释
- 程序的注释不是程序的一部分，不会对程序运行产生影响

程序的注释

- 多行注释：用 `/* */`

```
/* mp3 解码程序
   author : Guo Wei
   programmed on 2011.5.18
*/
int main()
{
    int bitrate; /* 比特率, 以 Kbps为单位 */
    int size;    /* 以字节为单位 */
    .....
}
```

程序的注释

- 单行注释：用 //

```
int main()  
{  
    int bitrate; // 比特率, 以 Kbps为单位  
    int size;    // 以字节为单位  
    .....  
}
```

程序文件的保存

- 本课程中的所有程序，都应该存为后缀名为 “.cpp” 的文件然后编译，不能存成 “.c” 文件！



北京大学
PEKING UNIVERSITY

信息科学技术学院

变量和数据类型 初探



美国加州1号公路Big Sur

什么是变量

- 变量就是一个代号，程序运行时系统会自动为变量分配内存空间，于是变量就代表了系统分配的那片内存空间，对变量的访问，就是对其代表的内存空间的访问。

什么是变量

- 变量就是一个代号，程序运行时系统会自动为变量分配内存空间，于是**变量就代表了系统分配的那片内存空间**，对变量的访问，就是对其代表的内存空间的访问。
- **变量有名字和类型两种属性**，不同变量的名字就对应了内存中的不同地址（即不同位置），而变量的类型，决定了一个变量占用多少个字节。

什么是变量

- 变量就是一个代号，程序运行时系统会自动为变量分配内存空间，于是**变量就代表了系统分配的那片内存空间**，对变量的访问，就是对其代表的内存空间的访问。
- 变量有名字和类型两种属性，不同变量的名字就对应了内存中的不同地址（即不同位置），而**变量的类型，决定了一个变量占用多少个字节**。变量名相当于房间号，类型相当于户型。
- 在C++语言中，变量要先定义，然后才能使用。“使用”有时也称为“引用”。读取或修改一个变量的值，都叫使用这个变量。**定义变量的语句，要出现所有使用该变量的所有语句之前。**

变量的定义

类型名 变量名1,变量名2,.....,变量名n;

```
int number, price; //定义整型变量 number和price
```

变量的定义

类型名 变量名1,变量名2,.....,变量名n;

```
int number, price; //定义整型变量 number和price
```

一个变量不能定义两次:

```
int number;  
double number; //错, 前面已经有定义了
```

变量的命名规则

- 变量由大小写字母、数字和下划线构成,中间不能有空格, 长度不限, 不能以数字开头。

- 合法变量名示例:

```
name    _doorNum    x1    y    z    a2    A    number_of_students  
PrintValue    MYTYPE
```

- 变量名是大小写相关的, name 和 Name 是不同的两个变量。

C++的保留字

- 变量名不能和C++系统预留的一些名字 (保留字)重复。C++预留的保留字为:

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
bool	catch	class	const_cast
delete	dynamic_cast	explicit	false
friend	inline	namespace	new
operator	private	protected	public
reinterpret_cast	static_cast	template	
this	throw	true	try
typeid	typename	using	virtual

此外还要注意不要取名为 begin、end、next、index、list、link 等

C++的数据类型

“数据类型”能够说明一个变量表示什么样的数据(整数, 实数, 还是字符等)。不同数据类型的变量, 占用的存储空间大小不同。
除了基本数据类型外, C++还允许程序员自定义数据类型。

C++的基本数据类型

- int,long,short,char的最高位(最左边)是符号位, 为1则表示负数, 为0表示非负数。
- 一个字节等于8个比特, 即8个二进制位

类型名	含义	字节数	取值范围
int	整型,表示整数	4	$-2^{31} \sim 2^{31}-1$
long	长整型, 表示整数	4	$-2^{31} \sim 2^{31}-1$
short	短整型, 表示整数	2	$-2^{15} \sim 2^{15}-1$
unsigned int	无符号整型, 表示非负整数	4	$0 \sim 2^{32}-1$
unsigned long	无符号长整型, 表示非负整数	4	$0 \sim 2^{32}-1$
unsigned short	无符号短整型, 表示非负整数	2	$0 \sim 2^{16}-1$
long long	64位整型, 表示整数	8	$-2^{63} \sim 2^{63}-1$
unsigned long long	无符号64位整型, 表示非负整数	8	$0 \sim 2^{64}-1$
float	单精度实数型, 表示实数	4	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
double	双精度实数型, 表示实数	8	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
char	字符型, 表示字符	1	-128 ~ 127
unsigned char	无符号字符型	1	0 ~ 255
bool	布尔类型, 表示真假	一般是1	true 或false

用sizeof运算符求变量占用字节数

sizeof(变量名)

sizeof(类型名)

能够得到某个变量或某一类型变量占用的字节数

```
int n1 = 10;  
double f;  
char c;  
printf("%d,%d,%d,%d", sizeof(n1), sizeof(short),  
        sizeof(double), sizeof(c));
```

输出: 4,2,8,1

变量的初始化

- 变量在定义的时候，可以给它指定一个初始值，这叫变量的初始化。

```
int a = 4, b = 3, c;  
char ch = 'a';
```

- 没有初始化的变量，其值一般是不确定的。



北京大学
PEKING UNIVERSITY

信息科学技术学院

变量和数据类型 进阶



张掖五彩丹霞

有符号整数和无符号整数

- short、int、long、long long 类型的变量，可以表示正数，也可以表示负数，称为有符号的整数类型。
- unsigned short, unsigned int, unsigned long, unsigned long long 类型的变量，只会被看作非负数，称为无符号的整数类型。

有符号整数的表示方式

- 将最左边的位（最高位）看作“符号位”。
符号位为0，则表示是非负数，其绝对值就是除符号位以外的部分；符号位为1，则表示是负数，其绝对值是所有位取反（0变1，1变0）后再加1。
- 将一个负整数表示为二进制的方法：
 - 1) 设置符号位为1
 - 2) 其余位等于绝对值取反再加1

有符号整数的表示方式

- 给定一个负整数的二进制表示形式，求该负整数：

该负整数的绝对值是其二进制表示形式取反再加1(取反加1后的结果要看作是正数)。

- 将一个负整数表示为二进制的方法：
 - 1) 设置符号位为1
 - 2) 其余位等于该负数的绝对值的二进制表示形式取反再加1

有符号整数的表示方式

整数	表示成short时的二进制形式	十六进制形式
0	0000 0000 0000 0000	0000
1	0000 0000 0000 0001	0001
257	0000 0001 0000 0001	0101
32767	0111 1111 1111 1111	7FFF
-32768	1000 0000 0000 0000	8000
-1	1111 1111 1111 1111	FFFF
-2	1111 1111 1111 1110	FFFE
-257	1111 1110 1111 1111	FEFF

数据类型的自动转换

- 有些不同的数据类型之间是相容的，可以互相赋值，这叫数据类型的自动转换

```
int a = 11.34;    // 11.34被自动转换为11后赋值给a
int b = 30;
double d = b;     // d的值是30.0
```

字符类型到整型的互相转换

- 字符型数据可以转换成整型数据

```
int k = 'a'; //k内容变为'a'的ASCII码, 即97  
printf("%d",k); //输出: 97
```

- 整型数据也可以转换为字符型数据, 但只会留下最右边的一个字节 (第0位到第7位), 其他字节丢弃

```
int n = 98;  
char k = n; //k内容变98,98是字符'b'的ASCII码  
printf("%c",k); //输出: b
```

类型自动转换示例

```
#include <stdio>
#include <iostream>
using namespace std;
int main() {
    int n1 = 1378;    //1378的十六进制形式是 0x562
    short n2;
    char c = 'a';
    double d1 = 7.809;
    double d2;
    n2 = c+1;         //n2变为98 , 97是'a'的ASCII码
    printf("c=%c,n2=%d\n",c,n2); //输出 c=a,n2=98
    c = n1;           // n1是0x562, 0x62被当做ASCII码赋值给c,c变为 'b'
    printf("c=%c,n1=%d\n",c,n1); //输出 c=b,n1=1378
    n1 = d1;          // d1=7.809, 去掉小数部分后赋值给n1,n1变为7
    printf("n1=%d\n", n1);    //输出 n1=7
    d2 = n1;          //d2变为7
    printf("d2=%f\n",d2);    //输出 d2=7.000000
    return 0;
}
```

c=a,n2=98
c=b,n1=1378
n1=7
d2=7.000000



北京大学
PEKING UNIVERSITY

信息科学技术学院

常 量



冰岛杰古沙龙冰河湖

什么是常量

- 常量就是在程序运行过程中值不会发生改变，而且一眼就能看出其值的量

12 'a' 7.809 "Hello,C++"

- 常量也可以分成多种
整型，浮点型，字符型，字符串,符号常量

整型常量

- 十进制整型常量

0 123 -456 677363

- 十六进制整型常量，以 “0x” 开头

0x123 -0x1a 0x2abcdef 0xFFA 0x100

十六进制数中，用A(a)表示10, B(b)表示11 F(f)表示15

$$(3450)_{10} = 0 \times 10^0 + 5 \times 10^1 + 4 \times 10^2 + 3 \times 10^3$$

$$0x2FAB = 11 \times 16^0 + 10 \times 16^1 + 15 \times 16^2 + 2 \times 16^3 = (12203)_{10}$$

一个十六进制位，正好对应于4个二进制位 $F = (1111)_2$

整型常量

- 八进制整型常量，以0开头

01 0123 -0456 0677

字符型常量

- 字符型常量表示一个字符，用单引号括起来

`'a' 'B' ',' '0' '9' ':'`

- 字符型常量可用于给char 和unsigned char类型的变量赋值

```
char c = 'a';  
unsigned char ch;  
ch = '9';
```

字符型常量

- 字符型常量和变量都占**一个字节**，内部存放的是字符的ASCII编码。
ASCII编码是一个0~255的整数

字符	二进制编码	十六进制编码	十进制编码
'A'	0100 0001	0x41	65
'a'	0110 0001	0x61	97
'0'	0011 0000	0x30	48
'!'	0010 0001	0x21	33

字母和数字的ASCII 十进制编码：

'0'~'9': 48 ~ 57

'A'~'Z': 65 ~ 90

'a'~'z': 97 ~ 122

字符型常量

- C++中还有一类字符常量，以'\'开头，如'\n'、'\r'、'\t'等，称为“转义字符”。所谓“转义”就是指'\'后面的字符被转成别的含义。

转义字符	含义	ASCII码
<code>\n</code>	换行，将输出位置移到下一行开头	10
<code>\r</code>	回车，将输出位置移到本行开头	13
<code>\t</code>	制表符，输出位置跳到下一个制表位置。制表位置一般是8的倍数加1	9
<code>\b</code>	退格，输出位置回退一个字符	8
<code>\\</code>	反斜杠 “\”	92
<code>\'</code>	单引号 “'”	39
<code>\0</code>	0字符	0
<code>\ddd(如 \123)</code>	ddd是个八进制数，代表字符的ASCII码	ddd(八进制)
<code>\xhh(如\x61)</code>	hh是个十六进制数，代表字符的ASCII码	hh(十六进制)

字符串常量

- 字符串常量是用双引号括起来的一串字符

"a" "abc" "1234567"

- "" 也是一个字符串常量，它代表一个空串，即不包含任何字符的字符串
- "a" 和 'a' 是不一样的，前者是只有一个字符的字符串，后者是一个字符，不能用前者给一个char类型的变量赋值。
- "1234567"当然也和1234567是不一样的，不能用前者给一个int类型变量赋值。

字符串常量

- 字符串常量里可以包含转义字符

```
printf("123\t456\nabc\n");  
printf("123\'45\n");  
printf("UVWX\"YZ\n");
```

123	456
abc	
123'45	
UVWX"YZ	

符号常量

- 为了阅读和修改的方便，常用一个由字母和数字组成的符号来代表某个常量，这样的常量就叫符号常量

#define 常量名 常量值

- 定义之后，程序中所有出现“常量名”的地方，就等价于出现的是“常量值”
- “常量名”的命名规则和变量相同，“常量值”则写什么常量都可以

```
#define MAX_NUM 1000  
#define UNIVERSITY_NAME "Peking University"  
#define MYINT i = 5;
```

- 尽量少使用数值常量，二用符号常量替代它，这样便于修改