

第七课 进制转换

知识目标

- 学习二进制
- 二进制与十进制，八进制和十六进制相互转换
- 十进制与八进制、十六进制相互转换
- C++语言位运算符

二进制数

只由 0 和 1 两个数字组成的数字就叫做二进制；

例如：

1101001100011000

10010010011100110111

二进制与十进制相互转换

1. 将十进制整数转换为二进制整数

将被转换的十进制数反复除以 2，直到商为 0，再将所得的余数倒排列（从末位读起）就得到了对应的二进制数，简单的说就是除 2 倒取余。

(1) 十进制整数 12 转换为二进制整数



(2) 十进制整数 15 转换为二进制整数

2	15	
2	7	余数: 1
2	3	余数: 1
2	1	余数: 1
	0	余数: 1




$(15)_{10} = (1111)_2$

2. 十进制小数转换为二进制小数

将十进制小数不断乘以 2 取整数, 直到结果为 1 或达到保留小数位要求, 将得到的整数顺序排列就得到了二进制数, 简单的说就是乘以 2 取整。

(1) 十进制小数 0.25 转换为二进制小数


0.25	
X 2	
0.5	
X 2	整数: 0
1	
	整数: 1



$(0.25)_{10} = (0.01)_2$

(2) 十进制小数 0.375 转换为二进制

0.375	
X 2	
0.750	
X 2	整数: 0
1.500	
	整数: 1
0.500	
X 2	
1.000	
	整数: 1



$(0.375)_{10} = (0.011)_2$

(3) 十进制小数 0.57 转换为二进制小数，保留 4 位小数

$$\begin{array}{rcl}
 0.57 & & \\
 \times 2 & & \\
 \hline
 1.14 & \text{整数: } 1 & \\
 0.14 & & \\
 \times 2 & & \\
 \hline
 0.28 & \text{整数: } 0 & \\
 \times 2 & & \\
 \hline
 0.56 & \text{整数: } 0 & \\
 \times 2 & & \\
 \hline
 1.12 & \text{整数: } 1 &
 \end{array}
 \quad \downarrow \quad
 (0.57)_{10} = (0.1001)_2$$

(4) 十进制小数 0.32 转换为二进制小数，保留 4 位小数


$$\begin{array}{rcl}
 0.32 & & \\
 \times 2 & & \\
 \hline
 0.64 & \text{整数: } 0 & \\
 \times 2 & & \\
 \hline
 1.28 & \text{整数: } 1 & \\
 0.28 & & \\
 \times 2 & & \\
 \hline
 0.56 & \text{整数: } 0 & \\
 \times 2 & & \\
 \hline
 1.12 & \text{整数: } 1 &
 \end{array}
 \quad \downarrow \quad
 (0.32)_{10} = (0.0101)_2$$

(5) 十进制小数 12.25 转换为二进制小数

$$\begin{array}{rcl}
 2 \overline{) 12} & & \\
 2 \overline{) 6} & \text{余数: } 0 & \\
 2 \overline{) 3} & \text{余数: } 0 & \\
 2 \overline{) 1} & \text{余数: } 1 & \\
 0 & \text{余数: } 1 &
 \end{array}
 \quad \uparrow \quad
 \begin{array}{rcl}
 0.25 & & \\
 \times 2 & & \\
 \hline
 0.5 & \text{整数: } 0 & \\
 \times 2 & & \\
 \hline
 1 & \text{整数: } 1 &
 \end{array}
 \quad \downarrow$$

$$(12.25)_{10} = (1100.01)_2$$

(6) 十进制小数 19.125 转换为二进制小数

$ \begin{array}{r} 2 \overline{) 19} \\ \underline{2 9} \\ 2 \overline{) 9} \quad \text{余数: 1} \\ \underline{2 4} \\ 2 \overline{) 4} \quad \text{余数: 1} \\ \underline{2 2} \\ 2 \overline{) 2} \quad \text{余数: 0} \\ \underline{2 1} \\ 2 \overline{) 1} \quad \text{余数: 0} \\ \underline{0} \quad \text{余数: 1} \end{array} $		$ \begin{array}{r} 0.125 \\ \times 2 \\ \hline 0.25 \quad \text{整数: 0} \\ \times 2 \\ \hline 0.5 \quad \text{整数: 0} \\ \times 2 \\ \hline 1 \quad \text{整数: 1} \end{array} $
---	---	---

$(19.125)_{10} = (10011.001)_2$

3. 二进制数转换为十进制数

首先将二进制数展开成多项式和的表达式, 然后将多项式按十进制逐项相加求和就得到了十进制数, 简单的说就是按权展开求和。

(1) 二进制数 101 转换为十进制数

$$\begin{aligned}
 (101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= 4 + 0 + 1 \\
 &= (5)_{10}
 \end{aligned}$$

(2) 二进制数 1111 转换为十进制数

$$\begin{aligned}
 (1111)_2 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 &= 8 + 4 + 2 + 1 \\
 &= (15)_{10}
 \end{aligned}$$

(3) 二进制小数 1011.11 转换为十进制数

$$\begin{aligned}
 (1011.11)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\
 &= 8 + 0 + 2 + 1 + 0.5 + 0.25 \\
 &= (11.75)_{10}
 \end{aligned}$$

(4) 二进制小数 101.001 转换为十进制数

$$\begin{aligned}
 (101.001)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 4 + 0 + 1 + 0 + 0 + 0.125 \\
 &= (5.125)_{10}
 \end{aligned}$$

4. 二进制数转换为八进制数

从低位到高位每 3 位一组，直接用八进制表示就可以了，简称三合一法。

(1) 二进制数 1011010011 转换为八进制数

$$\begin{array}{lcl}
 (1011010011)_2 & \xrightarrow{\text{分组}} & 001 \ 011 \ 010 \ 011 \\
 & \xrightarrow{\text{转换}} & 1 \quad 3 \quad 2 \quad 3 \\
 & \xrightarrow{\text{结果}} & (1323)_8
 \end{array}$$

(2) 二进制数 10110.0011 转换为八进制数

$$\begin{array}{lcl}
 (10110.0011)_2 & \xrightarrow{\text{分组}} & 010 \ 110 \ . \ 001 \ 100 \\
 & \xrightarrow{\text{转换}} & 2 \quad 6 \ . \ 1 \quad 4 \\
 & \xrightarrow{\text{结果}} & (26.14)_8
 \end{array}$$

(3) 二进制数 0.1 转换为八进制数

$$\begin{array}{lcl}
 (0.1)_2 & \xrightarrow{\text{分组}} & 001 \ . \ 100 \\
 & \xrightarrow{\text{转换}} & 1 \ . \ 4 \\
 & \xrightarrow{\text{结果}} & (0.4)_8
 \end{array}$$

5. 八进制数转换为二进制数

将八进制按照从左到右，每位展开为三位，小数点位置不变，简称一分三法。

(1) 将八进制 67.54 转换为二进制

$$\begin{array}{lcl}
 (67.54)_8 & \xrightarrow{\text{分组}} & 6 \quad 7 \ . \ 5 \quad 4 \\
 & \xrightarrow{\text{转换}} & 110 \ 111 \ . \ 101 \ 100 \\
 & \xrightarrow{\text{结果}} & (110111.101100)_2
 \end{array}$$

最低位的0可以删除掉

6. 二进制转换为十六进制

从低位到高位每 4 位一组，直接用十六进制表示就可以了，简称四合一法。

二进制 101111 转换为十六进制数

$$\begin{array}{lcl}
 (101111)_2 & \xrightarrow{\text{分组}} & 0010 \ 1111 \\
 & \xrightarrow{\text{转换}} & 2 \quad F \\
 & \xrightarrow{\text{结果}} & (2F)_{16}
 \end{array}$$

7. 十六进制转换为二进制

将十六进制按照从左到右，每位展开为四位，小数点位置不变，简称一分四法。

将十六进制数 AF 转换为二进制

$$\begin{array}{lcl}
 (AF)_{16} & \xrightarrow{\text{分组}} & A \quad F \\
 & \xrightarrow{\text{转换}} & 1010 \ 1111 \\
 & \xrightarrow{\text{结果}} & (10101111)_2
 \end{array}$$

8. 八进制转换为十六进制

八进制数 → 二进制数 → 十六进制数

将八进制数 67.54 转换为十六进制

$$\begin{array}{lcl}
 (67.54)_8 & \xrightarrow{\text{拆分}} & 6 \quad 7 \quad . \quad 5 \quad 4 \\
 & \xrightarrow{\text{转换}} & 110 \ 111 \ . \ 101 \ 100 \\
 & \xrightarrow{\text{中间值}} & (110111.101100)_2 \\
 & \xrightarrow{\text{拆分}} & 0011 \ 0111 \ . \ 1011 \\
 & \xrightarrow{\text{转换}} & 3 \quad 7 \quad . \ B \\
 & \xrightarrow{\text{结果}} & (37.B)_{16}
 \end{array}$$

9. 各种进制数的表示法

进制位	二进制	八进制	十进制	十六进制
规则	逢2进1	逢8进1	逢10进1	逢16进1
基数	2	8	10	16
数符	0 1	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9 A B C D E F
位权	2的1次方	8的1次方	10的1次方	16的1次方
形式表示	B	O	D	H

十进制与八进制，十六进制相互转换

1. 八进制转换为十进制

将八进制数 23 转换为十进制

$$\begin{aligned}
 (23)_8 &= 2 \times 8^1 + 3 \times 8^0 \\
 &= 16 + 3 \\
 &= (19)_{10}
 \end{aligned}$$

2. 十进制转换为八进制

(1) 十进制数 19 转换为八进制

$$\begin{array}{r|l}
 8 & 19 \\
 \hline
 8 & 2 \\
 \hline
 & 0
 \end{array}
 \begin{array}{l}
 \text{余数: } 3 \\
 \text{余数: } 2
 \end{array}
 \uparrow$$

$$(19)_{10} = (23)_8$$

3. 十六进制转换为十进制

十六进制数 2A 转换为十进制

$$\begin{aligned}
 (2A)_{16} &= 2 \times 16^1 + A \times 16^0 \\
 &= 32 + 10 \\
 &= (42)_{10}
 \end{aligned}$$

4. 十进制转换为十六进制

将十进制数 19 转换为十六进制

$$\begin{array}{r|l}
 16 & 19 \\
 \hline
 16 & 1 \\
 \hline
 & 0
 \end{array}
 \begin{array}{l}
 \text{余数: } 3 \\
 \text{余数: } 1
 \end{array}
 \uparrow$$

$$(19)_{10} = (13)_{16}$$

各进制转换总结

1. 进制数的表示方法

进位制	二进制	八进制	十进制	十六进制
规则	逢2进1	逢8进1	逢10进1	逢16进1
基数	2	8	10	16
数符	0 1	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9 A B C D E F
位权	2的一次方	8的一次方	10的一次方	16的一次方
形式表示	B	Q	D	H

2. 各进制数对应表

十进制	二进制	八进制	十六进制
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

3. 各种进制数之间的转换方法

- 二进制 -> 十进制
按位权乘 2 的 N-1 次方
- 十进制 -> 二进制
整数部分除 2 取余,
小数部分乘 2 取整
- 八进制 -> 十进制
按位权乘 8 的 N-1 次方
- 十进制 -> 八进制
整数部分除 8 取余,
小数部分乘 8 取整
- 十六进制 -> 十进制
按位权乘 16 的 N-1 次方
- 十进制 -> 十六进制
整数部分除 16 取余,
小数部分乘 16 取整
- 二进制 -> 八进制
每三位二进制数表示一位八进制数
- 八进制 -> 二进制
每一位八进制数表示三位二进制数
- 二进制 -> 十六进制
每四位二进制数表示一位十六进制数
- 十六进制 -> 二进制
每一位十六进制数表示四位二进制数
- 八进制 -> 十六进制
以二进制为中介,
先将八进制转换为二进制,
然后将二进制转换为十六进制

C++语言位运算符

1. 位运算符简介

位运算是指按二进制进行的运算。在系统软件中，常常需要处理二进制位的问题。C++语言位运算符包括以下几种：

- 与 (&)、或 (|)
- 异或 (^)、取反 (~)
- 左移 (<<) 与右移 (>>)

2. 位运算符：与 (&)

& 按位与，如果两个相应的二进制位都为 1，则该位的结果值为 1，否则为 0

例如：13&5

13 的二进制数为： 0000 1101

5 的二进制数为： 0000 0101

两个二进制数按位与： 0000 0101

所以 13&5 = 5;

```
#include<iostream>
using namespace std;
int main(){
    int a=13;
    int b=5;
    cout << (a&b) << endl;
    return 0;
}
```

3. 位运算符：或 (|)

| 按位或，两个相应的二进制位中只要有一个为 1，该位的结果值为 1。借用逻辑学中或运算的话来说就是，一真为真。

例如：60₍₈₎ | 17₍₈₎

60₍₈₎ 的二进制数为： 0011 0000

17₍₈₎ 的二进制数为： 0000 1111

两个二进制数按位与： 0011 1111

所以 60₍₈₎ | 17₍₈₎ = 63₍₁₀₎ ;

```
#include<iostream>
using namespace std;
int main(){
    int a=060;
    int b=017;
    cout << (a|b) << endl;
    return 0;
}
```

4. 位运算符：异或 (^)

^ 按位异或，若参加运算的两个二进制位值相同则为 0，否则为 1。

例如：3 ^ 5

3 的二进制数为： 0000 0011

5 的二进制数为： 0000 0101

两个二进制数按位与： 0000 0110

所以 3 ^ 5 = 6 ;

```
#include<iostream>
using namespace std;
int main(){
    int a=3;
    int b=5;
    cout << (a^b) << endl;
    return 0;
}
```

5. 位运算符：取反 (~)

~取反是一元运算符，用于求整数的二进制反码，即分别将操作数各二进制位上的 1 变为 0，0 变为 1。

例如：~21

15 的二进制数为： 0001 0101

~21 的二进制数为： 1110 1010

所以~21= -22 ;

```
#include<iostream>
using namespace std;
int main(){
    int a = 21;
    cout << (~a) << endl;
    return 0;
}
```

6. 位运算符：左移 (<<)

<< 左移运算符是用来将一个数的各二进制位左移若干位,移动的位数由右操作数指定(右操作数必须是非负值),其右边空出的位用 0 填补,高位左移溢出则舍弃该高位。

例如: $15 \ll 2$

15 的二进制数为: 0000 1111

左移两位后二进制数为: 0011 1100

所以 $15 \ll 2 = 60$;

```
#include<iostream>
using namespace std;
int main(){
    int a = 15;
    cout << (a<<2) << endl;
    return 0;
}
```

7. 位运算符：右移 (>>)

>>右移运算符是用来将一个数的各二进制位右移若干位,移动的位数由右操作数指定(右操作数必须是非负值),移到右端的低位被舍弃

例如: $15 \gg 2$

15 的二进制数为: 0000 1111

右移两位后二进制数为: 0000 0011

所以 $15 \gg 2 = 3$;

```
#include<iostream>
using namespace std;
int main(){
    int a = 15;
    cout << (a>>2) << endl;
    return 0;
}
```