

第 09 课 数学专题(二)

学习目标

编程知识

熟悉：原码、反码、补码

掌握：位运算

掌握：数学函数

拓展知识

计算机中的数据存储

数学函数等价表示

★ 重难点解析

场景重现：

你还记得在 C++ 中原码、反码、补码的含义吗？它们之间如何转换？

技能小贴士：

程序中的所有数值在计算机内存中都是以二进制的形式存储的。

转化成二进制后，选择最左侧的，也称最高位的位置表示符号位，其中 0 表示正数，1 表示负数

原码：

正数：符号位为 0，十进制数转化成二进制数

负数：符号位为 1，十进制数转化成二进制数

例如：13 的原码为 00001101，-13 的原码为 10001101

反码：

正数：与原码相同

负数：在原码的基础上，除符号位外取反（0 变 1，1 变 0）

例如：13 的反码为 00001101，-13 的反码为 11110010

补码：

正数：与原码，反码相同

负数：在反码的基础上加 1

例如：13 的补码为 00001101，-13 的补码为 11110011

★ 重难点解析

场景重现：

你还记得在 C++ 中有哪些位运算吗？位运算的结果如何计算？

技能小贴士：

位运算：直接对整数在内存中的二进制位进行按位操作。

按位与 $\&$

按位或 $|$

按位异或 \wedge

按位取反 \sim

左移 $<<$

右移 $>>$



小码术语箱

按位与 $\&$ 运算规则：全 1 才 1，否则为 0

$0 \& 0 = 0$ $0 \& 1 = 0$ $1 \& 0 = 0$ $1 \& 1 = 1$

按位或 $|$ 运算规则：全 0 才 0，否则为 1

$0 | 0 = 0$ $0 | 1 = 1$ $1 | 0 = 1$ $1 | 1 = 1$

按位异或 \wedge 运算规则：相异为 1，相同为 0

$0 \wedge 0 = 0$ $0 \wedge 1 = 1$ $1 \wedge 0 = 1$ $1 \wedge 1 = 0$

按位左移 $<<$ 运算规则：

1. 将整数转化成二进制数

2. 将所有数值向左移动，最高数值位丢弃，最低位补 0

按位右移 $>>$ 运算规则：

1. 将整数转化成二进制数

2. 将所有数值向右移动，最低位丢弃，最高数值位补 0

【例 1】

请问以下哪个选项是 $13 \& 3$ 的结果?

- A. 13
- B. 6
- C. 1
- D. 15

【思路分析】:

```
3: 0 0 1 1
13: 1 1 0 1
-----
    0 0 0 1
(0001)2 = 1
```

【答案】: C

【例 2】

请问以下哪个选项是 $13 | 11$ 的结果?

- A. 13
- B. 6
- C. 1
- D. 15

【思路分析】:

```
11: 1 0 1 1
13: 1 1 0 1
-----
    1 1 1 1
(1111)2 = 15
```

【答案】: D

【例 3】

请问以下哪个选项是 $13 \wedge 11$ 的结果?

- A. 13
- B. 6
- C. 1
- D. 15

【思路分析】:

11: 1 0 1 1

13: 1 1 0 1

0 1 1 0

$(0110)_2 = 6$

【答案】: B

【例 4】

请问以下哪个选项是 $13 \ll 2$ 的结果?

- A. 13
- B. 52
- C. 25
- D. 26

【思路分析】:

13: 0 0 1 1 0 1

1 1 0 1 0 0

$(110100)_2 = 52$

【答案】: B

【例 5】

请问以下哪个选项是 $13 \gg 2$ 的结果?

- A. 0
- B. 13
- C. 6
- D. 3

【思路分析】:

13: 0 0 1 1 0 1

 0 0 0 0 1 1 0 1
 $(000011)_2 = 3$

【答案】: D

★ 重难点解析

场景重现：

你还记得在 C++ 中有哪些数学函数吗？它们的运算结果是什么含义？

技能小贴士：

<code>pow(a, b)</code>	a^b	a 的 b 次方
<code>sqrt(a)</code>	\sqrt{a}	a 的二次方根
<code>abs(a)</code>	$ a $	a 的绝对值
<code>ceil(a)</code>	$\lceil a \rceil$	a 向上取整
<code>floor(a)</code>	$\lfloor a \rfloor$	a 向下取整

使用数学函数时，需要添加头文件 `#include <cmath>`

【例 6】

请问下列哪条语句是输出 6 的 3 次方？

- A. `cout << pow(6, 3);`
- B. `cout << pow(3, 6);`
- C. `cout << 3*3*3*3*3*3;`

【思路分析】：

`pow(a, b)`：a 是底数，b 是指数，计算结果是 a^b

【答案】：A

【例 7】

输入一个正整数 n ，输出 n 的算术平方根，并保留两位小数

输入格式：输入一个正整数 n ，($0 \leq n \leq 10000$)

输出格式：输出这个数的算术平方根，并保留两位小数

输入样例：100

输出样例：10.00

【思路分析】：

- 1、定义一个整型变量 n
- 2、输入 n
- 3、通过 `sqrt` 函数计算 n 的算术平方根
- 4、通过 `printf` 将结果保留两位小数输出

【代码】：

```
#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;
int main (){
    int n;
    cin >> n;
    printf("%.2lf",sqrt(n));
}
```


【例 8】

阅读以下代码，判断正确输出结果是什么？

```
#include <iostream>
#include <cstdio>
using namespace std;
int main (){
    printf("%.1f\n",ceil(2.3));
    printf("%.1f\n",floor(3.8));
    return 0;
}
```

- | | | | |
|--------|--------|--------|--------|
| A. 2.0 | B. 3.0 | C. 3.0 | D. 4.0 |
| 3.0 | 3.0 | 4.0 | 3.0 |

【思路分析】：

ceil 是向上取整函数，floor 是向下取整函数



【答案】： B



挑 战 状

本人_____

决心通过查找资料，独立完成下列挑战并理解题目。

完成挑战，绝不抄袭。特立此状，以示决心。

在 8 位二进制补码中，10101011 表示的数是十进制下的()

- A. 43
- B. -85
- C. -43
- D. -84

一个字长为 8 位的整数的补码是 11111001，则它的原码是()

- A. 00000111
- B. 01111001
- C. 11111001
- D. 10000111

在 C++ 语言中,表达式 $23|2^5$ 的值是()

- A. 18
- B. 1
- C. 23
- D. 32

监督人：



Tips

 单词:

square root	/ˌskwer ˈru:t/	-平方根
power	/ˈpaʊər/	-幂, 乘方
absolute	/ˈæbsəlu:t/	-绝对的