

# 20220606-C++

---

## 1.学习内容

### 1.1 C++

#### C++位运算

&按位与

|按位或

~按位非

^按位异或

左移

右移

Mnist数据处理常见的一个例子（大小字节转换）

## 2.结果描述

## 1.学习内容

### 1.1 C++

#### C++位运算

以下运算符只能用于带符号或无符号的char，short，int，long类型

#### &按位与

当两个相应的二进制位都为1，则该位的结果为1。否则为0。

```

1  int i=3&5;
2  -->3=011
3  -->5=101
4  -->i=001=1
5  应用：要取一个整数的低字节，只需要将该整数与几个1按位与即可
6  a 00101100 10101100
7  b 00000000 11111111
8  a&b-->c 00000000 10101100

```

## |按位或

两个相应的二进制位只要有一个为1，该位的结果值为1.

```

1  int i=3|5;
2  -->3=011
3  -->5=101
4  -->i=111=7

```

## ~按位非

一元运算符。将相应的二进制位的1变为0，0变为1。

```

1  int i=~5;
2  -->5=101
3  -->i=010=2

```

## ^按位异或

若两个相应的二进制位值相同则为0，否则为1

```

1  int i=3&5;
2  -->3=011
3  -->5=101
4  -->i=110=6
5  应用：使特定位翻转，将其与1111进行异或运算；保留原值，与0000进行异或运算
6  a 00101100 10101100
7  b 00000000 11111111
8  a^b-->c 00101100 01010011
9  应用：交换两个值，不用临时变量
10 a 011
11 b 100
12 a=a^b=111
13 b=b^a=011
14 a=a^b=100

```

## 左移

将一个数的各二进制位左移若干位，其右边空出的位用0填补，高位左移溢出则舍弃该高位

```

1  a=15=00001111;
2  a<<2;
3  a=00111100

```

## 右移

将一个数的各二进制位右移若干位，移到右端的低位被舍弃，对于无符号数，高位补0；对于有符号数，某些机器用符号位填补左边空出的部分，某些机器则用0填补

```

1  a:1001011111101101
2  a>>1: 0100101111110110 （逻辑右移时）
3  a>>1: 1100101111110110 （算术右移时）

```

Mnist数据处理常见的一个例子（大小字节转换）

Little endian（小字节序、低字节序）：低位字节排放在内存的低地址端，高位字节排放在内存的高地址端。

Big endian（大字节序、高字节序）：高位字节排放在内存的低地址端，低位字节排放在内存的高地址端。

LE 04 03 02 01

BE 01 02 03 04

C++ | 复制代码

```
1  int ReversalInt(int nValue)
2  {
3      unsigned char cTemp1=nValue&255;
4      unsigned char cTemp2=(nValue>>8)&255;
5      unsigned char cTemp3=(nValue>>16)&255;
6      unsigned char cTemp4=(nValue>>24)&255;
7      //首先利用static_cast强制转换为int,变为4个字节
8      int result= static_cast<int>(cTemp1)<<24;
9      result+=static_cast<int>(cTemp2)<<16;
10     result+=static_cast<int>(cTemp3)<<8;
11     result+=cTemp4;
12
13     return result;
14 }
15 -----
16 uint32_t convert_to_little_endian(const unsigned char* bytes)
17 {
18     return(uint32_t)(
19         (bytes[0]<<24)|
20         (bytes[1]<<16)|
21         (bytes[2]<<8)|
22         (bytes[3]
23         )
24     )
25 }
```

## 2.结果描述

今天总算开始找到一点学习的状态，后面继续加油。