位运算优先级

运算符（优先级从上往下）	运算符说明及应用场景	结合性
() [] -> .	括号（函数等），数组，结构体指针变量的成员访问，普通结构体变量的成员访问	由左向右
! ~ ++ -- + -	逻辑非，按位取反，自增1，自减1，正号，负号	由右向左
* & (类型) sizeof	指针，取地址，强制类型转换，求占用空间大小	
* / %	乘，除，取模	由左向右
+ -	加，减	由左向右
<< >>	左移，右移	由左向右

指针相关运算符的优先级

```
uint8_t m = 5;
uint8_t *p = &m;
*p++;   // ++ 优先级高于 * ，所以先执行p++，再执行*，如果想要修改m的数值，应该写为 (*p)++;
```

指针相关运算符的优先级

```
typedef struct List
{
   struct List *prev;
   struct List *next;
} List;

typedef struct TempHumiSensor
{
   uint32_t id;
   uint8_t humi;
   float temp;
   List list;
} TempHumiSensor;

TempHumiSensor *sensor = FindTempHumiSensor();
List *l = &sensor->list;
```

- & 和 -> 都是对sensor操作，->优先级高于&，所以先执行sensor->list，再执行&取地址

指针相关运算符的优先级

```
#define OFFSET_OF(typeName, memberName) \
        ((uint32_t)&((typeName *)0)->memberName)
```

- (uint32_t) & ((typeName *) 0) -> memberName

1. ((typeName *) 0) -> memberName， () 和 -> 优先级最高，根据结合性 由左向右，先执行()，将0强转为地址；
2. 再执行((typeName *) 0) -> memberName，访问成员memberName；
3. (uint32_t) & ((typeName *) 0) -> memberName， (uint32_t)是数据类型转换运算符，和 & 优先级相同，根据结合性，由右向左，先执行&，对成员memberName取地址；
4. 最后执行(uint32_t) & ((typeName *) 0) -> memberName，将地址值转换为普通数值。

THANK YOU!