**移位操作方法总结**

**1.获得int型最大值**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getMaxInt(){
2. **return**  (1<<31) - 1; //2147483647， 由于优先级关系，括号不可省略
3. }

**另一种写法**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getMaxInt(){
2. **return**  ~(1<<31); //2147483647
3. }

**C语言中不知道int占几个字节时候**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getMaxInt(){
2. **return**  ((unsigned  **int** )- 1 ) >>  1 ; //2147483647
3. }

**2.获得int型最小值**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getMinInt(){
2. **return**  1<<31; //-2147483648
3. }

**3.获得long类型的最大值**

**C语言版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **long**  getMaxLong(){
2. **return**  ((unsigned  **long** )-1) >> 1; //2147483647
3. }

**JAVA版**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **long**  getMaxLong(){
2. **return**  (( **long** ) 1 << 127 )- 1 ; //9223372036854775807
3. }

获得long最小值，和其他类型的最大值，最小值同理.

**4.乘以2运算**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  mulTwo( **int**  n){ //计算n\*2
2. **return**  n<<1;
3. }

**5.除以2运算**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  divTwo( **int**  n){ //负奇数的运算不可用
2. **return**  n>>1; //除以2
3. }

**6.乘以2的m次方**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  mulTwoPower(**int**  n, **int** m){ //计算n\*(2^m)
2. **return**  n<<m;
3. }

**7.除以2的m次方**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  divTwoPower(**int**  n, **int** m){ //计算n/(2^m)
2. **return**  n>>m;
3. }

**8.判断一个数的奇偶性**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **boolean**  isOddNumber(**int**  n){
2. **return**  (n &  1 ) ==  1 ;
3. }

**9.不用临时变量交换两个数（面试常考）**

**C语言 版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **void**  swap( **int**  \*a, **int**  \*b){
2. (\*a)^=(\*b)^=(\*a)^=(\*b);
3. }

**通用版（一些语言中得分开写）**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. a ^= b;
2. b ^= a;
3. a ^= b;

**10.取绝对值（某些机器上 ， 效率比n>0  ?  n:-n 高）**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  abs( **int**  n){
2. **return**  (n ^ (n >> 31)) - (n >> 31);
3. /\* n>>31 取得n的符号，若n为正数，n>>31等于0，若n为负数，n>>31等于-1
4. 若n为正数 n^0=n,数不变，若n为负数有n^-1 需要计算n和-1的补码，然后进行异或运算，
5. 结果n变号并且为n的绝对值减1，再减去-1就是绝对值 \*/
6. }

**11.取两个数的最大值（某些机器上， 效率比a>b ? a:b高）**

**通用版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  max( **int**  a, **int**  b){
2. **return**  b&((a-b)>>31) | a&(~(a-b)>>31);
3. /\*如果a>=b,(a-b)>>31为0，否则为-1\*/
4. }

**C语言版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  max( **int**  x, **int**  y){
2. **return**  x ^ ((x ^ y) & -(x < y));
3. 0101
4. 1010
5. 000
6. /\*如果x<y x<y返回1，否则返回0，
7. 、 与0做与运算结果为0，与-1做与运算结果不变\*/
8. }

**12.取两个数的最小值（某些机器上， 效率比a>b ? b:a高）**

**通用版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  min( **int**  a, **int**  b){
2. **return**  a&((a-b)>>31) | b&(~(a-b)>>31);
3. /\*如果a>=b,(a-b)>>31为0，否则为-1\*/
4. }

**C语言版**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  min( **int**  x, **int**  y){
2. **return**  y ^ ((x ^ y) & -(x < y));
3. /\*如果x<y x<y返回1，否则返回0，
4. 与0做与运算结果为0，与-1做与运算结果不变\*/
5. }

**13.判断符号是否相同**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **boolean**  isSameSign(**int**  x,  **int** y){
2. **return**  (x ^ y) >  0 ;  // true 表示 x和y有相同的符号， false表示x，y有相反的符号。
3. }

**14.计算2的n次方**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getFactorialofTwo(**int**  n){ //n > 0
2. **return**  2<<(n-1); //2的n次方
3. }

**15.判断一个数是不是2的幂**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **boolean**  isFactorialofTwo(**int**  n){
2. **return**  (n & (n -  1 )) ==  0 ;
3. /\*如果是2的幂，n一定是100... n-1就是1111....
4. 所以做与运算结果为0\*/
5. }

**16.对2的n次方取余**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  quyu( **int**  m, **int**  n){ //n为2的次方
2. **return**  m & (n -  1 );
3. /\*如果是2的幂，n一定是100... n-1就是1111....
4. 所以做与运算结果保留m在n范围的非0的位\*/
5. }

**17.求两个整数的平均值**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getAverage( **int**  x,  **int**  y){
2. **return**  (x+y) >>  1 ;
3. ｝

**另一种写法**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getAverage( **int**  x,  **int**  y){
2. **return**  ((x^y) >>  1 ) + (x&y);
3. /\*(x^y) >> 1得到x，y其中一个为1的位并除以2，
4. x&y得到x，y都为1的部分，加一起就是平均数了\*/
6. }

**下面是三个最基本对二进制位的操作**

**18.从低位到高位,取n的第m位**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  getBit( **int**  n,  **int**  m){
2. **return**  (n >> (m- 1 )) &  1 ;
3. }

**19.从低位到高位.将n的第m位置1**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  setBitToOne(**int**  n,  **int** m){
2. **return**  n | ( 1 <<(m- 1 ));
3. /\*将1左移m-1位找到第m位，得到000...1...000
4. n在和这个数做或运算\*/
5. }

**20.从低位到高位,将n的第m位置0**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **int**  setBitToZero(**int**  n,  **int** m){
2. **return**  n & ~( 1 <<(m- 1 ));
3. /\* 将1左移m-1位找到第m位，取反后变成111...0...1111
4. n再和这个数做与运算\*/
5. }

**另附一些对程序效率上没有实质提高的位运算技巧，一些也是位运算的常识（面试也许会遇到）**

**计算n+1**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. -~n

**计算n-1**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. ~-n

**取相反数**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. ~n +  1 ;

**另一种写法**

**[java]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. (n ^ - 1 ) + 1 ;

**if(x == a) x = b; if(x == b) x = a;**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. x = a ^ b ^ x;

**sign函数，参数为n，当n>0时候返回1，n<0时返回-1，n=0时返回0**

**[cpp]** [view plain](http://blog.csdn.net/nash_/article/details/8262185) [copy](http://blog.csdn.net/nash_/article/details/8262185)

1. **return**  !!n - (((unsigned)n>>31)<<1);