

第四章 函数

第45课 整数的位运算

《信息学奥赛一本通·编程启蒙 C++版》

运算符	含义	说明	例子
&	按位与	把参与运算的两个数对应的二进制位相与，只有对应的二进制均为1时，结果的对应位才为1，否则为0。	9&5 中 9 可以写成 (00001001), 5 可以写成 (00000101), 那么 9&5 的运算结果为 0000 0001, 输出结果是 1。
	按位或	把参与运算的两个数对应的二进制位相或，也就是只要对应的两个二进制位有一个为1时，其结果就为1。	9 5 相当于 00001001 00000101, 运算结果是 00001101, 输出结果是 13。
^	按位异或	把参与运算的两个数对应的二进制位相异或，当对应的二进制位上的数据不相同，结果对应为1时，否则为0。	$1^1=0, 1^0=1, 0^0=0, 0^1=1$ 9^5 相当于 00001001^00000101, 运算结果是 00001100, 输出结果是 12。
~	取反	把运算数的各个二进制位按位求反。	~9 相当于 ~(0000 1001), 运算结果为 1111 0110。
<<	左移	把“<<”左边的运算数的各二进制位向左移若干位，“<<”右边的数是指定移动的位数，高位丢弃，低位补0。	a<<4 指把 a 的各二进制位向左移动 4 位，如 a=00000011(十进制为 3), 左移 4 位后为 00110000(十进制 48)。
>>	右移	把“>>”左边的运算数的各二进制位全部右移若干位，“>>”右边的数是指定移动的位数。	设 a=15, a>>2 表示把 00001111 右移为 0000 0011(十进制为 3)。

【例 45.1】 高低位交换

【题目描述】

给出一个小于 2^{32} 的正整数。这个数可以用一个 32 位的二进制数表示（不足 32 位用 0 补足）。我们称这个二进制数的前 16 位为“高位”，后 16 位为“低位”。将它的高低位交换，我们可以得到一个新的数。试问这个新的数是多少（用十进制表示）。

例如，数 1314520 用二进制表示为 0000 0000 0001 0100 0000 1110 1101 1000（添加了 11 个前导 0 补足为 32 位），其中前 16 位为高位，即 0000 0000 0001 0100；后 16 位为低位，即 0000 1110 1101 1000。将它的高低位进行交换，我们得到了一个新的二进制数 0000 1110 1101 1000 0000 0000 0001 0100。它即是十进制的 249036820。

【输入格式】

一个小于 2^{32} 的正整数。

【输出格式】

一个整数表示交换后的数。

【样例输入】

1314520

【样例输出】

249036820

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. unsigned int n,num1,num2;
4. int main(){
5.     cin>>n;
6.     num1=(n<<16);
7.     num2=(n>>16);
8.     cout<<(num1|num2);
9.     return 0;
10.}
```

【例 45.2】 筷子大作战

【题目描述】

有个粗心的同学。今天在食堂打翻了洗碗大伯的篮子，筷子落了一地。

结果大伯怒了，硬是要这位同学把筷子一对一对配回去。

每只筷子都是有长度的，长度一样的筷子才算是一对。不过洗碗大伯跟他说其中有一只筷子是落单的。你能帮这位粗心的同学找出落单的筷子的长度吗？

【输入格式】

第一行为一个正整数 N ，代表筷子的支数 ($1 \leq N \leq 5000000$)。

接下去一行有 N 个正整数，代表每只筷子的长度 L_i ($1 \leq L_i \leq 2147483647$)，可能有很多筷子的长度都是一样的。

【输出格式】

输出落单的筷子的长度。

【样例输入】

5

1 2 1 2 3

【样例输出】

3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. long long n,x,ans;
4. int main(){
5.     ios::sync_with_stdio(0);
6.     cin>>n;
7.     for(int i=1;i<=n;i++){
8.         cin>>x;
9.         ans ^=x;
10.    }
11.    cout<<ans;
12.    return 0;
13.}
```

【例 45.3】 寻找最低数

【题目描述】

给你一个正整数 $A(1 \leq A \leq 2.1 \times 10^9)$ ，输出 A 的最低数。

例如，给你 $A=26$ ，我们可以将 A 化成二进制为 **11010**，则 A 的最低数是 **10**，输出 **10** 的十进制为 **2**。

再例如，给你 $A=88$ ，我们可以将 A 化成二进制为 **1011000**，则 A 的最低数是 **1000**，输出为 **8**。

【输入格式】

输入包含多组测试样例。每行输入一个正整数 $A(1 \leq A \leq 2.1 \times 10^9)$ 。当输入 **0** 时，输入结束。

【输出格式】

对于每一个输入，输出对应的最低数。

【样例输入】

26

88

0

【样例输出】

2

8

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int main(){
5.     while(cin>>n){
6.         if(n==0) break;
7.         cout<<(n&(-n))<<endl;
8.     }
9.     return 0;
10.}
```

【例 45.4】 数位翻转

【题目描述】

给定一个数 n ，你可以进行若干次操作，每次操作可以翻转 n 的二进制表示下的某一位，即将 0 变成 1 ， 1 变成 0 。

请问：至少需要多少次操作，才能将 n 变成 $n-1$ 。

【输入格式】

一个正整数 n 。 $1 < n \leq 10^9$

【输出格式】

输出最少的操作次数。

【样例输入】

10

【样例输出】

2

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int main(){
5.     cin>>n;
6.     n=n^(n-1);
7.     cout<<__builtin_popcount(n);
8.     return 0;
9. }
```

练 45.1 计算 2 的幂

【题目描述】

给定非负整数 n ，求 2^n ($1 \leq n \leq 31$)。

【输入格式】

输入只有一行，该行包含一个整数 n 。

【输出格式】

一个整数，即 2 的 n 次方。

【样例输入】

3

【样例输出】

8

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. int main(){
5.     cin>>n;
6.     cout<<(1<<n);
7.     return 0;
8. }
```

练 45.2 水题

【题目描述】

给定一个整数 n ，求出 $0 - n$ 之间满足 $x \text{ xor } (2 \times x) \text{ xor } (3 \times x) = 0$ 的整数 x 有多少个。

【输入格式】

输入只有一行，该行包含一个整数 n ($0 \leq n \leq 10^6$)。

【输出格式】

x 的个数

【样例输入】

2

【样例输出】

3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,cnt;
4. int main(){
5.     cin>>n;
6.     for(int i=0;i<=n;i++)
7.         if((i^(2*i)^(3*i))==0)cnt++;
8.     cout<<cnt;
9.     return 0;
10. }
```

练 45.3 小 B 的异或

【题目描述】

小 B 收到了一串数字，其中包含 n 个数字。

寄件人想知道这 n 个数的异或结果，但小 B 并不会求，就把这个问题转交给你。

但他为了使你求得的更方便，于是运用魔法把这 n 个数都变成了 1。

现在，你要求出这 n 个 1 异或后的结果。

关于异或，下表为 a 与 b 的异或结果：

a	b	$a \oplus b$
1	0	1
1	1	0
0	0	0
0	1	1

【输入格式】

输入只有一行，该行包含一个整数 n ($0 \leq n \leq 10^{18}$)。

【输出格式】

一行一个整数，为答案

【样例输入】

2

【样例输出】

0

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. long long n;
4. int main(){
5.     cin>>n;
6.     if(n%2==0) cout<<0;
7.     else cout<<1;
8.     return 0;
9. }
```

谢谢！

—