

L1212：汉明距离 ☆

题目描述

汉明距离是指**两个整数对应二进制位不同的数量**。

例如，如果 $x = 1$ (二进制表示为 0001)， $y = 4$ (二进制表示为 0100)，那么它们的汉明距离是 2。

输入输出格式

输入：一行两个数 x y ，表示需要计算的两个整数。其中 ($0 \leq x, y \leq 2^{30}$)

输出： x 和 y 的汉明距离。

输入示例	输出示例
1 4	2

样例解释

$$x = 1_{10} = 0001_2$$

$$y = 4_{10} = 0100_2$$

在二进制下， x 和 y 的表示如下：

位数	x	y	是否相同
第1位	1	0	不同
第2位	0	0	相同
第3位	0	1	不同
第4位	0	0	相同

x 和 y 二进制下有两处不同，所以汉明距离为 2。

背景知识

位运算是对整数在二进制表示下直接对位进行的操作：

- 按位与 ($\&$)：两位都为1时结果为1
- 按位或 ($|$)：两位至少有一个为1时结果为1
- 按位异或 ($^ / \oplus$)：两位不同时结果为1，相同时结果为0
- 左移 ($<<$)：将所有位向左移动，右边补0
- 右移 ($>>$)：将所有位向右移动，左边补0

算法分析：使用位运算计算汉明距离

1. 使用异或运算找出不同位

使用异或(\oplus)运算的特性：相同为0，不同为1

计算 $x \oplus y$ ，使得 x 和 y 不同的部分标记为1，相同的部分标记为0。

```
int t = x ^ y; // 计算异或结果
```

2. 使用与运算检查最后一位

使用 $\&$ 运算判断最后一位是否为1：

对于 $(t \& 1)$ ，如果值为1，则说明二进制下的 t 在最后一位是1（因为只有 $1\&1 = 1$ ）。

如果值为0，则说明二进制下的 t 在最后一位是0。

```
cnt += t & 1; // 检查最后一位是否为1
```

3. 使用右移运算逐位处理

使用右移($>>$)运算逐位统计1的个数：

通过循环处理最后一位，如果是1（意味着此处 x 和 y 不同）则计数器加1，然后将数字右移一位，直到数字变为0。

示例：计算 $x = 1, y = 4$ 的汉明距离

- 计算 $x \oplus y$ ：

$$x = 1_{10} = 0001_2$$

$$y = 4_{10} = 0100_2$$

$$x \oplus y = 0101_2$$

- 统计 $x \oplus y$ 中1的个数：

当前目标值	最后一位数值	计数器数值
0101	1 (x 和 y 不同)	1
010	0 (x 和 y 相同)	1
01	1 (x 和 y 不同)	2
0	0 (x 和 y 相同)	2

核心代码总览

```
int t = x ^ y; // 计算异或结果
int cnt = 0;    // 计数器
while (t) {     // 当目标值为0时终止
    cnt += t & 1; // 检查最后一位是否为1
    t >>= 1;       // 右移一位
}
```