

A 小L的位运算

题目描述

小L刚刚学习了位运算，还不是很明白，想从最简单的训练开始。他想知道两个整数的位与 $\&$ 、位或 $|$ 、位异或 \wedge 的结果。请你帮帮他。

输入

一行两个整数 a , b ，以空格分隔。

输出

一行，三个整数，分别为 a , b 的位与、位或、位异或的值，以空格分隔。

输入样例1

```
3 2
```

输出样例1

```
2 3 1
```

输入样例2

```
2147483647 1048575
```

输出样例2

```
1048575 2147483647 2146435072
```

数据范围

保证数据和结果均在`int`范围内。

AUTHOR: ljh

B 全额保险

题面描述

在五星宜居城市洛圣都，所有居民有很高额的保险。良好市民小夫在开车出门时发生了事故。保险公司为他赔付保费的上限为 a ，而他为本次事故需要花费 b 的修理费。小夫想知道保险公司的保费上限是否高于或等于修理费。

小夫还算有钱，低于 10^{-8} 的费用差异会被他认为是账目误差。所以当保费上限和修理费的差值低于此值，请告诉小夫两者基本上相等。

输入

输入一行两个浮点数 a ， b ，以空格分隔。

输出

两个浮点数基本相等时输出0，第一个浮点数大输出1，第二个浮点数大输出-1。

Hint

认定 a 和 b 基本相等时需要使用 `fabs(a-b) < 1e-8`，想想为什么？ 课件3-9题

使用 `fabs()` 需要在文件前面写 `#include <math.h>`

样例输入

样例1

```
0.000000000001 -0.000000000001
```

样例2

```
2 1
```

样例输出

样例1

```
0
```

样例2

1

数据范围

$-10000 \leq a, b \leq 10000$ 。

author: Yt

C 罗德厨房

题目描述

在罗德岛的课堂上，干员们刚刚学习了位运算相关的知识。但由于位运算如此之难，罗德厨房决定给干员们开小灶，为他们补充能量。今日的大厨给出了一份菜谱，并为第 i 种菜肴定下了初步价格 x_i

不过，在课堂上学的很好的 Exusiai 同学觉得这份定价并不合理，于是偷偷骇入罗德厨房的系统，将每种菜肴的价格的二进制表示下的第 **3** 位（ 2^2 位）设成了 0，而第 **1** 位（ 2^0 位）设成了 1。你能帮助 Exusiai 计算，这样调整后，所有菜肴的总价降低了多少吗？（若反而升高则输出负的升高值）

Hint

可以使用如下方法将数 x 的二进制第 **3** 位设成 0、第 **1** 位设成 1。

```
result = x & (~(1 << 2));    //第三位设为0
result = result | 1;         //第一位设为1
```

输入格式

- 第 1 行一个整数 N ，代表菜谱上的菜肴总数。
- 第 2 ~ $N + 1$ 行，每行一个整数 x_i ，代表第 i 道菜肴的初步价格。

输出格式

一行一个整数，代表所有菜肴的总价降低的值

输入样例 1

2
15
12

输出样例 1

7

输入样例 2

1
0

输出样例 2

-1

样例解释

第一个样例的两个价格 $15 = (1111)_2$ ， $12 = (1100)_2$ ，第 3 位设成 0、第 1 位设成 1 之后变成 $(1011)_2 = 11$ 和 $(1001)_2 = 9$ ，总价降低了 7。

数据范围

$$1 \leq N \leq 20, 1 \leq i \leq N, 0 \leq x_i \leq 10^3$$

Author: Red

D 第六感猜测

题目描述

小羊学姐听说上个星期举行了数模国赛，她对此很感兴趣。已知今年的数模国赛一共有 A 、 B 、 C 、 D 、 E 五道题目，每个参赛队伍有且仅能选择一道题目。在比赛统计数据出来之前，小羊学姐希望通过第六感提前猜测一下每道题目选择的队伍数量，她猜测 A 、 B 、 C 、 D 、 E 五道题目的选择队伍数量依次为 x_1 、 x_2 、 x_3 、 x_4 、 x_5 。比赛结束后，小羊学姐查询到 A 、 B 、 C 、 D 、 E 五道题目的

选择队伍数量依次为 y_1 、 y_2 、 y_3 、 y_4 、 y_5 。现在她想据此评估一下自己的第六感准确度怎么样，评估标准如下：

- 若猜测队伍数量与实际队伍数量相差小于100，则第六感准确度加3分；
- 若猜测队伍数量与实际队伍数量相差等于100，则第六感准确度加1分；
- 若猜测队伍数量与实际队伍数量相差大于100，则第六感准确度减1分*；

现在请你帮助小羊学姐的第六感准确度打分。【注意：可能猜测队伍数量多于实际数量，也可能猜测队伍数量少于实际数量，且两种情况都叫“相差”】

输入

一共两行输入，所有输入都是整数

第一行为 x_1 、 x_2 、 x_3 、 x_4 、 x_5

第二行为 y_1 、 y_2 、 y_3 、 y_4 、 y_5

输出

一共输出一个整数（可正可负）

Hint

参考例题【C2-10】，考查的是逻辑运算符的0或1结果的应用。

在判断相差时，可使用库函数abs()，需头文件stdlib.h，功能：求绝对值，用法类似fabs函数

输入样例

```
101 102 103 104 105
205 204 203 202 201
```

输出样例

```
5
```

样例解释

A 题猜测与实际相差大于100， B 题猜测与实际相差大于100， C 题猜测与实际相差等于100， D 题猜测与实际相差小于100， E 题猜测与实际相差小于100，则最终第六感准确度分数为 $-1 - 1 + 1 + 3 + 3 = 5$

数据范围

$100 \leq x_i, y_i \leq 1000$

E 翻转正整数

题目描述

相信大家都知道了计算机储存整数是二进制形式的，并且采用的是**补码**。对于一个小于 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**补齐）。我们称这个二进制数的前**16**位为“高位”，后**16**位为“低位”。我们称将这个二进制数的高低位交换为翻转。

例如十进制数4用二进制表示为：

$(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0100)$

其中前16位为高位，即 $(0000)(0000)(0000)(0000)$ ；

后16位为低位，即 $(0000)(0000)(0000)(0100)$ ，翻转后为：

$(0000)(0000)(0000)(0100)(0000)(0000)(0000)(0000)$ ，即262144。

输入格式

一个小于 2^{32} 的正整数。

输出格式

翻转后得到的正整数

输入样例

4

输出样例

262144

HINT

一个小于 2^{32} 的正整数可用`unsigned int`类型，其对应的格式字符为`%u`

读入无符号整数的示例如下：

```
unsigned int a;  
scanf("%u",&a);
```

位运算解此题不会？再给些提示：

(1) 取出整数`a`高位示例：`a&0xffff0000`，取出整数`a`低位示例：`a&0x0000ffff`

(2) 高位移至低位，低位移至高位 (`>>`，`<<`)

(3) 整合：用`|`还是`&`？

author:HiDen

F 词意分析

题目背景

在罗德岛，可露希尔正在教Lancet-2和Castle-3两个小机器人自然语言的处理。她将一些词告诉小机器人，并由他们处理成仅由 **0** 和 **1** 组成的 n 维语义向量。她希望你能帮助小车们检验识别的效果。

题目描述

语义向量的每一位被视作二进制的 **一个bit** 并给出对应的十进数（例如，将向量 $(1, 0, 1)$ 视作 **101** 并处理为 **5**）

你需要比较两个语义向量并输出两个向量的差异大小，即两向量之差的模长的平方。

对于 n 维向量 $\vec{a} = (a_0, a_1, a_2, \dots, a_n)$ ， a 的模长平方 $|\vec{a}|^2 = \vec{a}^2 = (a_0^2 + a_1^2 + a_2^2 + \dots + a_n^2)$ 。

输入格式

第一行一个整数 n ，代表词的数量。

接下来 n 行，每行两个十进制整数 a 、 b ，分别代表Lancet-2和Castle-3处理该词后输出的语义向量对应的数。

输出格式

n 行，每行一个整数代表两个语义向量的差异大小。

样例输入

```
2
7 5
13 43
```

样例输出

```
1
3
```

样例解释

$7 = (111)_2$ ，对应 $(1, 1, 1)$ ；

$5 = (101)_2$ ，对应 $(1, 0, 1)$ 。

两向量之差 = $(0, 1, 0)$ ，模长平方为1。

数据范围

$1 \leq n < 100, 0 < a, b < 2^{30}$

Hint

方法有很多，比如：

- 尝试将十进制转换为二进制并将每一位提取出来（参考：书P56的十进制转二进制）
- 使用刚学到的位运算符提取（参考： $(a >> 0) \& 1$ 提取了 a 的第0位， $(a >> 1) \& 1$ 提取了 a 的第1位）。

若使用数组保存，记得关注数据范围以选用合适的数组大小

在转换时别忘了高位还有默认的0。

Author : Red

G void学数学

题目描述

离散数学老师说：“我们有两个集合 A 和 B ,那么 A 和 B 的对称差就是 $(A - B) \cup (B - A)$ ”

比如 $A = 1, 2, 3, 4$ $B = 2, 5, 6$

那么 $A - B = 1, 3, 4$, $B - A = 5, 6$ 。

$A - B$ 就是将 A 中属于 B 的部分去掉
, $B - A$ 就是将 B 中属于 A 的部分去掉。

A 和 B 的对称差就是 $1, 3, 4, 5, 6$ ，是上面两个集合的并

然后老师就留下了 T 道课后作业，每一道都要求出 A 、 B 两个集合的对称差。
但是void很笨，他不会算，所以他想请你帮帮他。

输入

第一行一个整数 T ，代表作业的数量（数据组数）

对于每一组数据

第一行为一个整数 n ，表示集合 A 的基数，也就是集合 A 中有多少个元素

第二行有 n 个整数，表示集合 A 中的元素

第三行为一个整数 m ，表示集合 B 的基数

第四行有 m 个整数，表示集合 B 中的元素

输出

输出共 T 行，每一行从小到大输出若干个整数，整数之间以一个空格隔开，表示 A 和 B 的对称差中的所有元素

样例

输入

```
1
4
1 2 3 4
3
2 5 6
```

输出

```
1 3 4 5 6
```

数据范围

对于100%的数据， $1 \leq T \leq 1000$

对于60%的数据， $1 \leq n, m \leq 500$

对于100%的数据， $1 \leq n, m \leq 4000$

对于100%的数据， A, B 集合中的元素均在1~5000之间

并且保证 A, B 集合各自内部不会有重复的元素（也就是输入的 A, B 集合一定是合法的集合）

Hint

请仔细观察样例，对称差集合中每个元素在 A 集合和 B 集合里总共出现的次数，方法可参考课件例题【例3-10】

我们的所有元素范围只有1~5000，所以开一个多大的数组就能满足统计要求？

多组数据，那么对于每一组数据，我们首先要将这个数组初始化为0

author : void

H 蓝和橙的简单单位运算

题目描述

橙今天在寺子屋学习了基本的二进制知识和位运算法则，现在精通数学的八云蓝准备来考考她对这些知识点的掌握如何。

蓝写下了一个十进制整数交给橙，要求橙将这个数字的二进制码的某四位替换成另一个数字的二进制码。

但是橙并没有听的很明白，所以请你帮帮她回答这个问题。

输入格式

不定行输入

每一行，为三个十进制整数, num , n , key ,分别用一个空格分隔。

num 为待替换的数字， n 为需要替换的数位的最右一位的位置(以最右侧一位为第0位)， key 为替换在 num 上的值。

输出格式

不定行输出

对于每一行输入的数据，输出一行，为一个十进制整数 num' ，为替换之后的结果。

样例

输入样例

```
255 0 8
255 4 1
```

输出样例

```
248
31
```

样例说明

255 的二进制码为 11111111 ， 8 的二进制码为 1000, 1 的二进制码为 0001

将 255 的从第 0 位开始向左四位的二进制码替换为 1000 ，结果为 11111000 ，这是 248 的二进制码，因此输出 248

将 255 的从第 4 位开始向左四位的二进制码替换为 0001 ，结果为 00011111 ，这是 31 的二进制码，因此输出 31

数据范围

输入不超过 1000 行

$$16 \leq num \leq 2147483647$$

$$0 \leq key \leq 15 \text{ （即保证 } key \text{ 的二进制码不会超过 4 位）}$$

保证要替换的数位不会溢出 num 的位数。

Hint

八云紫给你传来了一张小纸条，上面写着：『参考ppt上的例题3-8：用C语言给一个数a的bit7~bit17赋值937，同时给bit21~bit25赋值17』

位运算符与赋值运算符的结合使用

例：用C语言给一个数a的bit7~bit17赋值937，同时给bit21~bit25赋值17

```
unsigned int a = 0xc305bad3;
a &= ~( ((1<<11) - 1) << 7 );
a |= 937<<7;
a &= ~( ((1<<5) - 1) <<21);
a |= 17<<21;
printf("a = 0x%x.\n", a);
```

1. $0x7ff$ 为 $0..0\ 0111\ 1111\ 1111$ ，即，初始化低11（十一）位为1， $((1<<11) - 1)$
2. $((1<<11) - 1) << 7$ ，
得到 $0011\ 1111\ 1111\ 1000\ 0000$
把第1步的十一个1左移7位（这十一个1变成bit7~bit17）
3. $\sim((1<<11) - 1) << 7$
bit7~bit17的十一个1变成0，其他位的0变成1，即变为 $1100\ 0000\ 0000\ 0111\ 1111$
4. $a \&= \sim(0x7ff << 7)$ ，保留a的其他位，但把a的bit7~bit17都置为0
5. $a |= (937 << 7)$ ，把a的bit7~bit17置为937
6. bit21~bit25赋值为17，原理同上

计算器	
程序员	
HEX	7FF
DEC	2,047
OCT	3 777
BIN	0111 1111 1111

Author: 星辰的微光

I 反码计算机

题目描述

Toby有一个与众不同的计算机，这个计算机用32位反码储存整数。

以十进制输入两个数字 a, b ，请计算这两个数字在Toby的于总不同的计算机上的按位与、按位或、按位异或的结果，并以十进制形式输出。

具体的，你需要将 a, b 的反码表示出来并带入位运算，进行位运算之后，结果应该也是反码的表示形式，请将结果反码换算成十进制后输出。

输入

一行两个十进制整数 a, b

输出

输出三行。

- 第一行一个十进制整数，为按位与的结果；
- 第二行一个十进制整数，为按位或的结果；
- 第三行一个十进制整数，为按位异或的结果。

输入样例1

1 2

输出样例1

0
3
3

输入样例2

-1 -2

输出样例2

-3
-0
3

样例解释1

1 的反码表示为 0000 0000 0000 0000 0000 0000 0000 0001
2 的反码表示为 0000 0000 0000 0000 0000 0000 0000 0010
按位与的结果为 0000 0000 0000 0000 0000 0000 0000 0000，表示 0
按位或的结果为 0000 0000 0000 0000 0000 0000 0000 0011，表示 3
按位异或的结果为 0000 0000 0000 0000 0000 0000 0000 0011，表示 3

样例解释2

-1 的反码表示为 1111 1111 1111 1111 1111 1111 1111 1110
-2 的反码表示为 1111 1111 1111 1111 1111 1111 1111 1101
按位与的结果为 1111 1111 1111 1111 1111 1111 1111 1100，表示 -3
按位或的结果为 1111 1111 1111 1111 1111 1111 1111 1111，表示 -0

按位异或的结果为 0000 0000 0000 0000 0000 0000 0000 0011，表示 3

数据范围

保证 $-2,147,483,647 \leq a, b \leq 2,147,483,647$ 且 a, b 可能是 -0

Hint

请注意考虑 -0 和 0 在反码中是有区别的

鸣谢

感谢各位老师对本题的指导

J 不可分解的01串

题目描述

求长度为 n 的 01 字符串中不可分解字符串的个数。

“不可分解”指不能写成两个或多个相同字符串的拼接。

- 100100100 == 100 * 3，可分解
- 1101，不可分解

由于结果可能很大，请输出答案对 998244353 取模的结果。

输入

一个正整数 n 。

输出

一行一个整数，表示答案对 998244353 取模的结果。

样例

输入#1

2

输出#1

2

输入#2

4

输出#2

12

数据范围

n 为正整数

对于95%的数据， $n \leq 2 \times 10^6$

对于5%的数据， $n \leq 10^{15}$

Author: 哪吒