

E3 - Statement

A 摩卡与补码 2

题目描述

补码是一种用来表示有符号整数的编码方式，广泛用于计算机系统中。与原码和反码不同，补码能够统一处理正数和负数，并且简化了计算机中的加减运算。在 C 语言中，有符号整数常是以 32 位补码表示的。

现在，输入一个 32 位二进制补码，请你输出它对应的有符号整数。

输入

多行输入，每行一个 32 位二进制补码。

保证输入不超过 10000 行。

输出

对于每行输入，输出一行，为对应补码表示的有符号整数。

输入样例

```
00000000000000000000000000000101
1111111111111111111111111111011
00000000000000000000000000000000
```

输出样例

```
5
-5
0
```

Hint

本题希望同学们能够对 `整数在计算机中存储的即为二进制补码形式` 有更深刻的了解，如果你感到这个任务很困难的话，不妨试试理解以下代码。

```
#include <stdio.h>

int main() {
    int num = 0;
    for(int i = 31 ; i >= 0 ;i--)
        num |= (1 << i);
    // num 现在的二进制表示是 11...1
    printf("%d", num);

    return 0;
}
```

每次读入**一位**整数（不仅仅局限于 01 位，而是 0 - 9 的数字位），可以参考如何代码

```
#include <stdio.h>

int main() {
    int bit = 0;
    scanf("%1d", &bit);
    printf("%d", bit);

    return 0;
}
```

Author : Moca

B 摩卡与冰雹序列

题目描述

Moca 最近了解到一个有趣的数学猜想，称为“冰雹猜想”。根据冰雹猜想，给定一个正整数 n ，对其进行如下操作：

- 如果这个数字是奇数，则将它乘以 3 再加 1；
- 如果这个数字是偶数，则将它除以 2。

无论从哪个正整数开始，经过若干次操作后，最终结果都会变成 1。例如，假设 $n = 6$ ，变化的过程为 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 。

Moca 想知道，对于给定的正整数 n ，经过这些操作后生成的序列是怎样的。

输入

一个正整数 n ，保证 $2 \leq n \leq 100000$ 。

输出

输出两行。

第一行一个正整数 k ，表示由描述中的操作生成的序列长度（包括初始值和 1）。

第二行 k 个整数，表示题目中描述的序列。

输入样例

6

输出样例

9
6 3 10 5 16 8 4 2 1

Hint

数组开多大合适呢，可以编程验证题目数据范围内形成的最长冰雹序列长度不超过 500（具体是多少可以编程验证一下）。

C 奇怪的位运算 2024

题目描述

有这样一种奇怪的位运算 \odot ，它的运算法则如下

p	q	$p \odot q$
0	0	0
0	1	0
1	0	1
1	1	0

你能迅速的给出两个数进行这种奇怪的位运算后的得数吗？

输入

多组数据输入，每组数据为一行两个正整数 a, b ， $0 \leq a, b \leq 10^{18}$ ，保证数据组数不超过 10^5

输出

对每组数据输出一行一个数字，为 $a \odot b$ 的值

输入样例

```
7 10
14 19
```

输出样例

```
5
12
```

样例解释

- 样例1: 7 的二进制为 00000111，10 的二进制为 00001010， $7 \odot 10$ 的二进制即为 00000101，也就是十进制的 5。
- 样例2: 14 的二进制为 00001110，19 的二进制为 00010011， $14 \odot 19$ 的二进制即为 00001100，也就是十进制的 12。

Hint

这题能不能不开数组呢 (-)

D 宁愿选择榴莲不放手

题目描述

「Zeratul」在母星「Shakuras」的石头巢穴里探险时，见到了一只十分喜爱榴莲的石猴（后人云：可石窝，有石猴，宁愿选择榴莲不放手~），他决定出一个问题考考这只石猴。

众所周知，一个榴莲内部有若干块果肉，计量单位常用“房”。现在「Zeratul」找到了 n 个榴莲，其中编号为 i 的榴莲有 A_i 房果肉。请你统计，对于任意正整数 x ，果肉为 x 房的榴莲有多少个。

输入

第一个数为榴莲个数 $n(1 \leq n \leq 10^6)$;
第二行, n 个由空格分隔的正整数 $A_1, ..., A_n(1 \leq A_i \leq 10000)$, 表示每个榴莲的房数。

输出

若干行, 输出格式为 $x : k$ 的格式, 其中 x 从小到大排序, k 表示房数为 x 的榴莲数, 如果 $k \neq 0$ 。
 $x, :, k$ 之间各有一个空格。

输入样例

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

输出样例

```
1 : 1
2 : 2
3 : 2
5 : 1
```

样例解释

房数为 1, 2, 3, 5 的榴莲数分别是 1, 2, 2, 1 。并且由于不存在 4 房的榴莲, 因此不输出 4 : 0 。
Author: Protoss

E Orch1d 的军训

题目描述

教官让同学们站成一排报数 (第 i 个同学报的数为 i) , 之后给出若干个**向后转**的口令, 有如下条件:

- 开始时同学们都**面向**教官;
- 报数为 1 的同学每次都要执行向后转口令;
- 如果报数为 i 的同学再次面向教官, 则报数为 $i + 1$ 的同学立刻向后转。

Orch1d 想知道, 在确定教官的口令数量 n 后, 最终哪些同学会**背对**教官。

输入

一个自然数 n , 表示教官的口令数量。

输出

第一行一个自然数 k , 表示背对教官的同学数量;
第二行 k 个升序排列的正整数, 表示背对教官的同学报的数, 用空格隔开。

输入样例1

```
3
```

输出样例1

```
2
1 2
```

输入样例2

```
10
```

输出样例2

```
2
2 4
```

数据范围

对于 30% 的数据，保证 $0 \leq n \leq 10^5$ ；

对于 100% 的数据，保证 $0 \leq n \leq 10^{18}$ 。

样例解释

对于输入样例 1，教官一共给出3次向后转的口令。

开始时同学们均面向教官；

第一次口令后，报数为 1 的同学背对教官；

第二次口令后，报数为 1 的同学再次面向教官，因此报数为 $1 + 1 = 2$ 的同学立刻向后转；

第三次口令后，报数为 1 的同学背对教官；

最后可以得到报数为 1 和 2 的同学背对教官，其余同学均面向教官，因此背对教官的同学数量为 2，背对教官的同学报的数为 1 和 2。

Hint

请聪明的你观察一下，这与刚学的进制有什么关系呢~

以及：不要把同学们转晕啊



F

与或异或



题目描述

小 P 获得了一个共有 n 个元素的有限数列，分别为 a_1, a_2, \dots, a_n 。小 P 将这个数列按顺序从左到右分为连续且非空的三段，并按照与、或、异或的顺序，求得每一段的价值。其中，第一段价值为按位与 ($\&$) 之和，第二段价值为按位或 ($|$) 之和，第三段价值为按位异或 (\oplus) 之和。小 P 想知道，这三段价值的加和（总价值）最大值为多少？

通俗来讲，你可以选取任意的 i, j ($1 \leq i < j < n$)，此时第一段为 a_1, a_2, \dots, a_i ，第二段为 $a_{i+1}, a_{i+2}, \dots, a_j$ ，第三段为 $a_{j+1}, a_{j+2}, \dots, a_n$ ，总价值为 $(a_1 \& a_2 \& \dots \& a_i) + (a_{i+1} | a_{i+2} | \dots | a_j) + (a_{j+1} \oplus a_{j+2} \oplus \dots \oplus a_n)$ 。求总价值的最大值。

输入格式

输入共两行。

第一行输入一个整数 n 。其中， $3 \leq n \leq 8000$ 。

第二行输入 n 个整数，分别为 a_1, a_2, \dots, a_n 。其中， $0 \leq a_i \leq 10^9$ 。

输出格式

输出共一行。

请输出总价值的最大值

样例输入1

```
5
2 2 6 4 9
```

样例输出1

```
21
```

样例解释

对于样例，取 $i = 2, j = 3$ ，此时第一段为 2, 2，第二段为 6，第三段为 4, 9。第一段价值为 2，第二段价值为 6，第三段价值为 13，总价值为 21。可以证明，此时没有其他的 i, j ，使总价值大于 21。

HINT

可以思考一下以下两组比较的答案：

比较 $a \& b$ 和 a 的大小；

比较 $a | b$ 和 a 的大小。

本题为搬运题，原题中 $3 \leq n \leq 3 \times 10^5$ ，其他不变。

author：小P

G 让我把记忆煎成饼（小数据）

题目描述

「James Eugene Raynor」让「Space Construction Vehicle」建造了一个煎饼摊，在「Hyperion」号上兜售煎饼，并投放广告：“再给我两根葱，让我把记忆煎成饼”。

和地球上的煎饼果子不同，为了迎合各种船员的口味，「Raynor」准备了 n 种配料，并按添加时间**从后到先**分别编号为 $1, 2, \dots, n$ ，第 i 中配料每份价格是 P_i 元。已知不添加任何配料的煎饼售价为 6 元，每种配料最多只添加一份。

在做一套煎饼时，「Raynor」可以选择一些配料添加到煎饼中，不过，每当添加一份配料，「Raynor」还会额外收取一部分钱当燃料费。具体地，在制作一套煎饼时，如果在所有添加的配料中，配料 i 是倒数第 j 份被添加到煎饼里的配料，那么「Raynor」对这份配料会收取 $P_i \times j$ 元的费用。

现在你让「Raynor」将所有可能的搭配都制作了一遍，请你算算你需要支付多少元。

输入格式

第一行，一个正整数 $n(n \leq 20)$ ，代表配料种数；

第二行， n 个由空格分开的正整数 $P_i(P_i \leq 10)$ ，含义如题目所示。

输出格式

一行，一个正整数，代表你需要支付的价钱。

样例输入

```
2
3 4
```

样例输出

```
42
```

样例解释

总共有 4 种煎饼，分别是不添加配料、添加配料 1、添加配料 2 和添加全部配料的。

- 不添加配料：底价 6 元，不加价，价格为 6 元；
- 添加配料 1：底价 6 元，配料 1 倒数第一个加入，收取 $3 \times 1 = 3$ 元，总价格为 9 元；
- 添加配料 2：底价 6 元，配料 2 倒数第一个加入，收取 $4 \times 1 = 4$ 元，总价格为 10 元；
- 添加全部配料：底价 6 元，配料 1 倒数第一个加入，收取 $3 \times 1 = 3$ 元，配料 2 倒数第二个加入，收取 $4 \times 2 = 8$ 元，总价格为 17 元。

因此你需要支付 $6 + 9 + 10 + 17 = 42$ 元。

Hint

由于每种配料只有加或不加两种可能，总共不超过 20 种配料，因此你可以尝试用一个整型变量表示一种煎饼，并用一个循环枚举所有可能的情况。例如，5 的二进制表示为 101，就可以表示添加了第一种和第三种配料，没加第二种配料的煎饼。

Author: Protoss

H 粗心的 Orch1d

题目背景

Orch1d 刚刚学会了不同进制之间的转换，但是他在整理答案时总是犯错误，你能帮助他改正吗？

题目描述

每当 Orch1d 将一个十进制数 N 转换成新的进制时，他总会写错一位数字。例如，将 14 转化成 2 进制数，正确的结果是 1110，但他可能会写成 0110 或 1010。Orch1d 不会增加或删减数字，但他可能会写出以 0 开头的错误数字。

现给出经 Orch1d 转换后的 N 的**二进制**形式和**K 进制**形式，请计算出 N 的正确数值（用**K 进制**表示）。

输入

第一行为整数 K ($3 \leq K \leq 16$);

第二行为 N 的二进制表示 (其中一位是错误数字) ;

第三行为 N 的 K 进制表示 (其中一位是错误数字) 。

输出

输出 N 的正确值 (用** K 进制**表示) 。

输入样例1

```
3
0010
111
```

输出样例1

```
101
```

输入样例2

```
16
11010011010001100100
DC564
```

输出样例2

```
D3564
```

数据范围

保证所有数据满足 $0 \leq N \leq 10^9$, 以及输出结果 N 的唯一性。

Hint

可以尝试更改每一位的值, 说不定就能找到正确结果OvO

如果大家在输入上遇到了问题, 可以尝试以下方法读入字符串并获取其长度:

```
#include<stdio.h>
#include<string.h>//处理字符串的库函数
int main()
{
    char s[105]; //定义长度为105的字符串
    scanf("%s",s); //读入字符串
    int len=strlen(s); //获取字符串s的长度
}
```

如果想要把输入的字符数组转化为十进制, 可以尝试以下方法:


```
//假设输入的字符数组为k进制，dec为转化后的十进制结果
dec=0;
for(int i=0;i<len;i++)
{
    if(s[i]>='0'&&s[i]<='9')
    {
        dec=dec*k+s[i]-'0';
    }
    else
    {
        dec=dec*k+s[i]-'A'+10;
    }
}
```

I 武魂融合技

题目背景

欢迎来到 BUAA 大陆，在这里生活的人都拥有属于自己的魂技，为了抵御强大的魂兽的侵略，人们修炼出了每两个人之间可以使用的武魂融合技，你能帮忙找出所有人中哪两个人的武魂融合技可以打出最高的伤害吗？

题目描述

每个人魂技的伤害可以用一个自然数表示。假设有两个人魂技的伤害分别为 a 和 b ，则他们武魂融合技的伤害值为 $a \& b$ ，其中 $\&$ 表示与运算。

给出 n 个人的魂技伤害值，请你找出所有可能的武魂融合技伤害的最大值。

输入

第一行一个正整数 n ，表示人数；

第二行 n 个自然数 $a_i (1 \leq i \leq n)$ ，表示每个人魂技的伤害值。

输出

一个自然数，表示武魂融合技伤害的最大值。

输入样例1

```
3
1 2 3
```

输出样例1

```
2
```

输入样例2

```
6
1 1 4 5 1 4
```

输出样例2

```
4
```

数据范围

对于 30% 的数据，保证 $2 \leq n \leq 100$;

对于 100% 的数据，保证 $2 \leq n \leq 10^5$, $0 \leq a_i \leq 2^{31} - 1$ 。

Author : With An Orch1d

J ddz 与位运算等式

题目描述

ddz 学完位运算之后在桌子上发现了一张草稿纸，上面的一个等式部分模糊不清了，隐约看见 $(xxx|114) \oplus (xxx\&514) = 1919810$ (其中 \oplus 表示按位异或， $|$ 表示按位或， $\&$ 表示按位与)，他只记得模糊的地方是两个相同的数字，ddz 想知道在知道其他三个数的情况下有多少个数字可以填进模糊处使得等式成立。

形式化来说，给出三个数 a, b, c ，问有多少个 d ，满足 $(d|a) \oplus (d\&b) = c$

输入

第一个数为数据组数 n ($0 < n \leq 50000$)。

接下来 n 行，每行三个整数 a, b, c ($0 \leq a, b, c < 2^{60}$)

输出

对于每组数据，输出一行，所有满足题意的 d 的数量

输入样例

```
2
5 9 6
9 5 6
```

输出样例

```
4
0
```

样例解释

- 1. 对于 $a = 5, b = 9, c = 6$ 来说满足条件的 d 有 3, 7, 11, 15。
- 2. 对于 $a = 9, b = 5, c = 6$ 来说没有满足条件的 d 。

Author : ddz

K max(a,b)

题目背景

通过一天25小时废寝忘食地敲代码，小 p 宣布他即将写出可以比肩 GCC 的编译器）。现在小 p 向你发出了内测邀请，但是小 p 并不知道你是否能提出有效的建议，所以他决定先给你出一道题。

题目描述

小 p 希望你在不进行数值比较的操作的情况下求出给定两个整数的较大值：具体来说，对于给出的每组整数 a, b , 计算 $\max(a, b)$ 。

你可能无法使用包括但不限于 `if`, `switch cases`, `while` 等语句、`qsort` 等函数。

输入

一共三组数据。

对于每组数据，一行两个整数，分别为 a 和 b 。（保证 a, b 在 `int` 范围内）

输出

对于每组数据，输出 a 和 b 的较大值。

输入样例

```
3 5
-1 -2
404 908
```

输出样例

```
5
-1
908
```

Hint

不要尝试进行任何**比较**操作（以及不要定义**奇怪的函数名或字符串**）

你可以采用下面的方式输入数据

```
int a, b;
scanf("%d%d", &a, &b);
// print answer
scanf("%d%d", &a, &b);
// print answer
scanf("%d%d", &a, &b);
// print answer
```