

# C4 - Statement

## A 摩卡与 lcm

### 题目描述

在第一次比赛中，我们使用了暴力枚举的方法来计算两个数的最大公约数（GCD），然后根据公式：

$$\text{lcm}(a, b) = \frac{a \times b}{\text{gcd}(a, b)}$$

计算两个数的最小公倍数（LCM）。

现在，我们已经掌握了更高效的方式来计算最大公约数。让我们再次挑战这个问题，通过更快的算法来求解两个数的最小公倍数。

### 输入格式

第一行，输入一个整数  $n$ ，表示数据的组数，保证  $1 \leq n \leq 10000$ 。

接下来  $n$  行，每行表示一组数据，包含两个正整数  $a$  和  $b$ ，保证  $1 \leq a, b \leq 10^9$ 。

### 输出格式

对应每行输入，输出一行整数，表示对应  $a$  和  $b$  的最小公倍数。

### 输入样例

```
2
12 15
7 5
```

### 输出样例

```
60
35
```

### Hint

$$\text{lcm}(a, b) = \frac{a \times b}{\text{gcd}(a, b)}$$

【例4-18】最大公约数 用辗转相除法求两个整数a和b的最大公约数。

```
#include <stdio.h>
int main()
{
    int a, b, r;
    scanf("%d%d", &a, &b);
    if(b == 0) {
        printf("gcd is: %d\n", a);
        return 0;
    }
    for(r = a%b; r != 0; r = a%b) {
        a = b;
        b = r;
    }
    printf("gcd is: %d\n", b<0 ? -b : b);
    return 0;
}
```

```
while((r = a%b) != 0) {
    a = b;
    b = r;
}
```

可看出，此时用while语句使得结构更清晰些？

```
for(; (r = a%b) != 0; ) {
    a = b;
    b = r;
}
```

注意：for循环头的三个表达式都可以省略，即写成for(;;)，此时第一个第三个表达式表示不执行操作，第二个表达式表示一个非0（即为真）的条件取值。这是死循环？如何退出？

B 庄园地图生成

题目描述

在《小僵尸吃脑子》的题目中，输入数据点是一张庄园地图，地图上只有空地、坚果和脑子。但出题人 Paradise 忘记了每个数据点的地图！不过好在，由于地图的规模最大是  $64 \times 64$ ，可以通过生成一系列随机数的方式来构造地图！

已知地图可以表示  $n$  行  $m$  列。Paradise 给了你  $n$  个  $m$  位二进制数的十进制值（一定是非负整数！），如果按顺序将这  $n$  个数的  $m$  位二进制表示写出来，每个数占一行，那么就可以得到一个  $n \times m$  的字符矩阵。如果某个位置的字符是 0，那么这个位置就是空地；如果某个位置的字符是 1，那么这个位置就是坚果

现在请你输出这张地图，用 . 表示空地，用 # 表示坚果。

输入

第一行，两个由空格分开的正整数  $n, m (n, m \leq 64)$ ，表示地图大小  
接下来  $n$  行，每行一个自然数  $A_i (A_i \leq 2^{64} - 1)$ ，表示地图第  $i$  行的十进制表示

输出

共  $n$  行，每行  $m$  个字符 . 或 #，表示地图

输入样例

```
2 3
2
6
```

# 输出样例

```
.#.
##.
```

# 样例解释

2 的 3 位二进制表示是 010，表示地图第一行是 `.#.`  
6 的 3 位二进制表示是 110，表示地图第二行是 `##.`

# Hint

有一种数据类型可以存储本题范围的数据—— `unsigned long long`，你可以使用 `scanf("%llu", &a);` 来读入一个 `unsigned long long` 类型的数据。

# C 摩卡与成绩统计 3

## 题目描述

Moca 需要对参与免修考试的同学们的成绩进行统计。

比赛由  $A - K$  共 11 道题组成，其中  $K$  题为诚信说明；除了  $K$  题外，其余十道题的过题数与比赛得分的折算规则如下：

过题数	1	2	3	4	5	6	7	8	9	10
比赛得分	15	30	45	60	75	90	95	99	100	100

如果  $K$  题未通过，比赛得分 60 分及以上记为总分 60 分；比赛得分 60 分以下记为总分 0 分；如果  $K$  题通过，则直接将比赛得分记为总分。

当给出一个学生的入学年份  $year$  和每道题的通过情况  $pass_i$ ，请按以下要求输出：

- 对于 2024 级新生，若总分  $score \geq 90$ ，则为通过免修考试，输出 `Congratulations, you pass the exam and your score is xx.` 其中 `xx` 为对应的总分；否则输出 `You have to take the course.`
- 对于非 2024 级新生，若总分  $score \geq 60$ ，则为通过免修考试，输出 `Congratulations, you pass the exam and your score is xx.` 其中 `xx` 为对应的总分；否则输出 `You have to take the course.`

## 输入

第一行一个整数  $n$ ，表示参与免修考试的学生数，保证  $1 \leq n \leq 50000$ 。

接下来  $n$  行，每行 12 个整数；第一个整数  $year_i$  表示第  $i$  个学生的入学年份，保证  $2021 \leq year_i \leq 2024$ ，后面十一个整数  $pass_{ij}$  依次对应第  $i$  个学生  $A - K$  题的通过情况，0 代表未通过该题，1 代表通过该题。

## 输出

输出  $n$  行，对每个学生，按要求输出一行对应的字符串。

## 输入样例

```
4
2024 1 1 1 1 1 0 1 1 1 1 1
2023 1 1 1 1 1 0 0 0 0 0 1
2024 1 1 1 1 1 0 0 0 0 0 1
2023 1 1 1 1 1 0 1 1 1 1 0
```

## 输出样例

```
Congratulations, you pass the exam and your score is 100.
Congratulations, you pass the exam and your score is 75.
You have to take the course.
Congratulations, you pass the exam and your score is 60.
```

Author: Moca

# D 小懒獭与日期

## 题目描述

小懒獭最近经常忘记今天是星期几，导致她不知道该按时去上什么课。

为了帮助小懒獭，Moca 决定编写一个程序，根据输入的日期，准确地告诉小懒獭今天是星期几。

## 输入

不定组输入，每组输入一行，每行三个正整数  $year$  ,  $month$  和  $day$  , 分别代表日期的年月日; 保证  $1600 \leq year \leq 3000$  ,  $1 \leq month \leq 12$  ,  $1 \leq day \leq 31$  ; 保证输入的日期是一个合法日期 (即不会出现 `2000 2 31` 这样不合法日期的)

保证输入不超过 10000 行。

## 输出

对于每行输入，计算出对应的日期是星期几后，输出对应的字符串：

- 星期一：输出 `Monday`
- 星期二：输出 `Tuesday`
- 星期三：输出 `Wednesday`
- 星期四：输出 `Thursday`
- 星期五：输出 `Friday`
- 星期六：输出 `Saturday`
- 星期日：输出 `Sunday`

## 输入样例

```
2024 10 20
2024 10 22
2004 5 1
```

## 输出样例

```
Sunday
Tuesday
Saturday
```

## Hint

可以参考 PPT 上给出的 Zeller 公式（详细参考例 4 - 26）哦。

**注意：** Zeller 公式只适合求 1582 年 10 月 10 日之后的情景，如果月份为 1、2 月份，那么要相当于前一年的 13、14 月份计算（详见下图中的 **note**）

# Zeller公式：计算任意一天是星期几

$$W = \left( \left\lfloor \frac{C}{4} \right\rfloor - 2C + Y + \left\lfloor \frac{Y}{4} \right\rfloor + \left\lfloor \frac{26(M + 1)}{10} \right\rfloor + D - 1 \right) \bmod 7$$

Where

$W$ : the day of week. (0 = Sunday, 1 = Monday, ..., 5 = Friday, 6 = Saturday)

$C$ : the zero-based century. (=  $\lfloor \text{year}/100 \rfloor = \text{century} - 1$ )

$Y$ : the year of the century. (=  $\begin{cases} \text{year} \bmod 100, & M = 3, 4, \dots, 12, \\ (\text{year} - 1) \bmod 100, & M = 13, 14. \end{cases}$ )

$M$ : the month. (3 = March, 4 = April, 5 = May, ..., 14 = February)

$D$ : the day of the month.

**NOTE:** In this formula January and February are counted as **months 13 and 14 of the previous year**. E.g. if it is 2010/02/02, the formula counts the date as 2009/14/02.

Author: Moca

## E 寻找因数 (easy version)

### 题目描述

$a$  是  $b$  的因数，当且仅当  $b \bmod a = 0$ ，其中  $\bmod$  代表求模运算。

Gino 给了你一个正整数，他想知道这个正整数的正因数一共有多少个，这些因数分别是多少，请你帮帮他吧！

### 输入

第一行一个正整数  $t$  代表数据组数， $1 \leq t \leq 10$ 。

接下来  $t$  行，每行一个正整数  $n$ ， $1 \leq n \leq 10^{12}$ 。

### 输出

对于每组数据，第一行输出正整数  $n$  的因数的个数  $m$ 。第二行输出  $m$  个正整数，为  $n$  的所有因数。要求从小到大输出。

### 输入样例

```
3
12
60
41
```

# 输出样例

```
6
1 2 3 4 6 12
12
1 2 3 4 5 6 10 12 15 20 30 60
2
1 41
```

# 样例解释

12 的因数有 6 个，从小到大分别是 1, 2, 3, 4, 6, 12

60 的因数有 12 个，从小到大分别是 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60

41 是质数，其因数只有 1 和它本身。

Author: Gino

# F 小水獭与 AC

## 题目描述

课程即将过半，难度也越来越大。为了给你加油鼓气，13 只小水獭排列成了 AC 的图案：

```
 *   ***
***  *
*   * ***
```

热爱编程的你很好奇，假如有更多的水獭加入进来，这个图案将变成什么呢？

具体的说，如果我们把 A 对应的图案的第一行中的水獭数记为  $2k - 1$ ，那么 A 共有  $2k + 1$  行，第  $k + 1$  行有  $4k - 1$  只水獭，其余各行每行只有两只水獭（这里说的只是图案 A 中的水獭），最后一行中左面的水獭前没有空格；其余各行第一只水獭前均有若干空格，以保证 A 的两条边斜度为 1。

对于图案 c，也有  $2k + 1$  行，其中 c 中最后一行最左面的水獭与 A 中最后一行最右面的水獭中间有一个空格；c 除了第一行和最后一行中有  $4k - 1$  只水獭，其余各行均只有一只水獭，每行的第一只水獭在垂直方向上对齐。

现在输入  $k$ ，请你输出对应的图案。

## 输入

一个正整数  $k$ ，含义如背景所示，保证  $1 \leq k \leq 100$ 。

## 输出

若干行字符串，表示  $k$  对应的 AC 图案，水獭用 \* 表示。

## 输入样例 1

```
1
```

## 输出样例 1

```
 *   ***
***  *
*   * ***
```

输入样例 2

2

输出样例 2

```

***      *****
 *    *    *
*****    *
 *          *
 *          *
 *          * *****
```

输入样例 3

3

输出样例 3

```

*****      *****
 *    *    *
 *          *
*****      *
 *          *
 *          *
 *          * *****
```

Hint

可以将原图案分为：第 1 行；第 2 至  $k$  行；第  $k + 1$  行；第  $k + 2$  至  $2k$  行；第  $2k + 1$  行五部分分别输出。

Author: Moca

G 大数变小数

题目描述

$ddz$  有一个数  $x$ ，他可以对这个数进行如下操作：将  $x$  替换成  $x$  的各位数字之和。 $ddz$  想知道进行  $k$  次操作后  $x$  的值是多少。

输入

第一行两个数  $n, k$ ， $n$  表示  $x$  的位数， $k$  表示操作次数 ( $1 \leq n \leq 10^6, 0 \leq k < 10^{18}$ )

第二行，一个数  $x$  ( $0 \leq x < 10^{10^6}$ )。

输出

输出一个整数，为对  $x$  进行  $k$  次操作后的值。

输入样例（1）

3 2  
999





# 输出样例

```
4
0
```

Author: ddz

# I 进入虚空

## 题目描述

《乌尔纳的低语》  
吾之智慧，将是你的传承。  
吾之传承，铸就你的未来。  
吾之未来，终将归于虚空.....

「James Eugene Raynor」、「Sarah Louise Kerrigan」和「Artanis」率领三族部队来到了「Ulnar」——通向虚空的大门，但这扇巨型传送门必须要特定的密码才能激活。

传送门会输出  $n$  个正整数  $A_1, A_2, \cdots, A_n$ ， 和一个正整数  $m$ ， 而密码则是满足  $1 \leq i < j < k \leq n$ ， 且  $A_i + A_j + A_k = m$  的三元组  $(i, j, k)$  的个数。

现在请你编写程序，帮助他们计算出密码。

## 输入格式

第一行， 两个由空格隔开的正整数  $n(3 \leq n \leq 3000), m(1 \leq m \leq 10^9)$ ；  
第二行，  $n$  个由空格隔开的正整数  $A_i(1 \leq A_i \leq 10^9)$ 。保证这  $n$  个正整数单调递增，即  $\forall i \in 1, 2, \cdots, n-1, A_{i+1} \geq A_i$ 。

## 输出格式

一行， 一个自然数， 表示密码。

## 样例输入

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

## 样例输出

```
4
```

## 样例解释

符合题意的三元组为  $(1, 2, 5), (1, 3, 5), (1, 4, 5), (2, 3, 4)$  共 4 组。

Author: Protoss

# J 超级数组加密极速版

## 题目描述

有一串长度为  $n$  的数  $a_1, a_2, \dots, a_n$ ,  $0 \leq a_i \leq 10^8$ , 它们组成一个数组。出于某种原因, Gino 想把这个数组加密, 于是他找来两位操作员, 让他们分别将原数组  $a$  按照以下方式**加密数组任意正整数次**:

- 任选一个数字  $i$ ,  $2 \leq i \leq n - 1$ , 花费  $2^i$  元, 将  $a_i$  加上 2, 将  $a_{i-1}$  和  $a_{i+1}$  都减去 1

可是, 其中一位操作员听错了 Gino 的要求, 他会按照这个方式来**加密数组任意正整数次**:

- 任选一个数字  $i$ ,  $2 \leq i \leq n - 1$ , 花费  $1^i$  元, 将  $a_i$  加上 3, 将  $a_{i-1}$  减去 2, 将  $a_{i+1}$  减去 1

请你根据两位操作员加密完毕后得到的数组, 计算两位操作员分别花费了多少元。

注意你并不要求出两名操作员的操作次数。

数据保证两位操作员都进行了不少于 1 次操作, 不多于  $10^8$  次操作。

## 输入

第一行一个正整数  $t$ , 代表数据组数,  $1 \leq t \leq 10^4$ 。

对于每组数据, 第一行一个正整数  $n$ , 代表数组长度,  $5 \leq n \leq 10^5$ 。

接下来两行, 每行  $n$  个整数  $a_1, a_2, \dots, a_n$ ,  $-9 \times 10^8 \leq a_i \leq 9 \times 10^8$ , 代表两位操作员分别完成加密操作后得到的两个加密后数组。

数据保证  $\sum n \leq 3 \times 10^5$ , 即所有数据的  $n$  的总和不超过  $3 \times 10^5$ 。

数据保证, 两位操作员得出的数组是由同一个原始数组通过以上操作得到的。

## 输出

对于每组数据, 输出一行, 包含一个字符串  $s$  和两个整数  $u, v$ , 它们之间都用一个空格分隔。

如果第一个操作员操作正确, 那么  $s$  是 `oui`, 否则  $s$  是 `non`。

此外,  $u, v$  分别对应是操作员 1 和操作员 2 花费的金额。如果操作正确, 那么你需要输出金额除以 1234567 的余数, 否则输出金额除以 7654321 的余数。

## 输入样例

```
2
5
0 3 4 3 5
1 0 6 3 5
5
-1 2 3 0 1
0 3 -1 3 0
```

## 输出样例

```
oui 12 1
non 2 20
```

## 样例解释

对于第一组样例, 原始数组为  $\{1, 2, 3, 4, 5\}$ , 操作员 1 正确加密了 2 次, 花费了 12 元, 操作员 2 错误加密了 1 次, 花费了 1 元。

对于第二组样例，原始数组为 {1, 1, 1, 1, 1}，操作员 1 错误加密了 2 次，花费了 2 元，操作员 2 正确加密了 2 次，花费了 20 元。

# 提示

- 「E2 - 2024级程序设计基础第二次练习赛」 「J 数组加密」
- 「Special - 2024国庆思维训练特别赛」 「BC 超级数组加密国庆版」

---

Author: Gino, Saisyc