

PCS-J 2022 模拟赛

题目概况

中文题目名称	小 H 做算术	小 H 打比赛	小 H 拼字符串	小 H 玩游戏
英文题目名称	calculate	contest	string	game
可执行文件名	calculate	contest	string	game
输入文件名	calculate.in	contest.in	string.in	game.in
输出文件名	calculate.out	contest.out	string.out	game.out
每个测试点时限	1s	1s	1s	1s
运行内存上限	512 MB	512 MB	512 MB	512 MB
结果比较方式	全文比较	全文比较	SPJ	全文比较
题目类型	传统	传统	传统	传统

编译选项：

语言	编译命令
C++	-lm -Wl,--stack=2147483647 --std=c++14 -O2

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
4. 选手应将各题的源程序放在选手文件夹内，不要建立子文件夹。

小 H 做算术 (calculate)

题目描述

小 H 正在做一道算术题，但是这道题的计算量太大了，于是他向你求助。

给定正整数 x, n, k, P ，请求出下式的结果。

$$(x^{0k} + x^{1k} + x^{2k} + \dots + x^{nk}) \bmod P$$

其中 x^n 是幂次运算，代表 n 个 x 相乘的结果。例如： $10^5 = 10 \times 10 \times 10 \times 10 \times 10 = 100000$ 。

$x \bmod n$ 代表取余运算，当进行带余除法 $x \div n = q \dots r$ 时， r 被称为余数，也即 $x \bmod n$ 的结果。例如： $14 \div 4 = 3 \dots 2$ ，于是 $14 \bmod 4 = 2$ 。

输入格式

第一行输入四个正整数 x, n, k, P 含义如题所述。

输出格式

一行输出一个非负整数表示答案。

样例输入 1

```
5 2 3 17
```

样例输出 1

```
9
```

$x = 5, n = 2, k = 3, P = 17$ ，需求出

$$(5^{0 \cdot 3} + 5^{1 \cdot 3} + 5^{2 \cdot 3}) \bmod 17$$

的结果，也就是 $(1 + 125 + 15625) \bmod 17 = 15751 \bmod 17 = 9$ 。

数据范围与约定

对于 100% 的数据，满足如下条件。

- $1 \leq x \leq 10^9$,
- $1 \leq n \leq 10^6$,
- $1 \leq k \leq 10^6$,
- $1 \leq P \leq 10^9$ 。

对于 20% 的数据，满足 $x = 1$ 。

对于另外 20% 的数据，满足 $n, k \leq 10^2$ 。

对于另外 20% 的数据，满足 $n, k \leq 10^3$ 。

小 H 打比赛 (contest)

题目描述

小 H 参加了一场比赛，一共有 n 位选手（包括小 H）参加了本场比赛，编号为 $1, 2, \dots, n$ 。

在本场比赛前，每一位选手的积分都是已知的，编号为 i 的选手在比赛前积分为正整数 a_i 。

现在比赛的最终排行榜还没有公布，但是小 H 事先了解到，在本场比赛中排名第 i 位的选手的积分将被加上整数 b_i ，本场比赛中不会出现并列的排名，保证对于 $1 \leq i < n$, $b_i \geq b_{i+1}$ 。

注意 b_i 可能为 0 或负数，代表该选手发挥不佳，积分不变或者下降。注意积分下降后可能变为 0 或者负数。

现在小 H 想要知道，对于每一位选手，是否存在一个本场比赛的排行榜，使得在积分更新后这位选手的积分在所有选手中排名最高？比较两位选手时，先比较他们的积分。若积分不同则积分大者排名较高。否则积分相同，比较他们的编号，编号小者排名较高。

输入格式

第一行一个正整数 n 表示参加比赛的人数。

第二行 n 个正整数 a_1, a_2, \dots, a_n 分别表示 n 位选手在比赛前的积分。

第三行 n 个整数 b_1, b_2, \dots, b_n 分别表示排名 i 位的选手将获得的分数奖励。

输出格式

n 行每行一个字符串，若编号为 i 的选手能够成为排名最高的选手，输出一行一个字符串 `win`，否则输出一行一个字符串 `Lose`（注意大小写）。

样例输入 1

```
5
3 1 6 4 7
4 2 1 0 -3
```

样例输出 1

```
win
Lose
win
win
win
```

对于编号为 1 的选手，若排行榜从第 1 名到第 5 名依次为 1, 4, 3, 5, 2，那么最终 5 位选手的分数分别为 7, -2, 7, 6, 7，虽然有三位选手获得了 7 分的最高积分，但是由于 1 号选手是其中编号最小的，他的排名也将是最高。

数据范围与约定

对于 100% 的数据，满足如下条件：

- $2 \leq n \leq 2 \times 10^3$,
- 对于 $i \in [1, n]$, $1 \leq a_i \leq 10^4$ 。
- 对于 $i \in [1, n]$, $-10^4 \leq b_i \leq 10^4$ 。

对于 30% 的数据, 满足 $n \leq 10$ 。

对于另外 30% 的数据, 满足 $n \leq 500$ 。

小 H 拼字符串 (string)

题目描述

小 H 收到了有三个长度相等的，只由小写英文字母 ($a \sim z$) 组成的字符串 A, B, C ，设 $n = |A| = |B| = |C|$ ，其中 $|A|$ 为 A 的长度。

可惜小 H 对这些字符串不是很满意，他希望构造出一个最美丽的字符串。

由于小 H 是数学爱好者，他这样定义一个字符串的美丽度：

对于字符串 S ，定义它的美丽度 $f(S)$ 为下式：

$$\left(\sum_{i=1}^{|S|} \text{ord}(S_i) \cdot x^i \right) \bmod P$$

其中 S_i 表示 S 从左到右第 i 位上的字符， $\text{ord}(c)$ 表示小写英文字母 c 在 26 个字母中排行第几，例如 f 排行第 6，于是 $\text{ord}(f) = 6$ 。

小 H 只可以构造出长度为 n 的字符串 S ，并且还需要满足对于 $i \in [1, n]$ ， $S_i = A_i$ 或者 $S_i = B_i$ 或者 $S_i = C_i$ 。

现在小 H 想要找到他能够构造的字符串中，美丽度最大的一个。

输入格式

第一行三个正整数 n, x, P ，含义如上所述。

接下来三行每行一个长度为 n ，仅由小写英文字母组成的字符串 A, B, C 。

输出格式

第一行一个非负整数表示小 H 能够得到的最大美丽度。

第二行一个长度为 n 的字符串表示最大的美丽度所对应的，小 H 能够构造出的字符串。若有多个可以构造出的字符串，请输出任意一个。

样例输入 1

```
2 10 1000
ab
ba
ac
```

样例输出 1

```
320
bc
```

由于 $\text{ord}(b) = 2, \text{ord}(c) = 3$ ， $f(bc) = (2 \cdot 10^1 + 3 \cdot 10^2) \bmod 1000 = 320$ 。

数据范围与约定

对于 100% 的数据，满足以下要求：

- $1 \leq n \leq 28$ 。
- $2 \leq x < P \leq 10^9$ 。

对于 40% 的数据, 满足 $1 \leq n \leq 10$ 。

小 H 玩游戏 (game)

题目描述

小 H 有一个长度为 n 的正整序列 a_1, a_2, \dots, a_n 。现在他准备和他的两个好朋友小 L 和小 J 用这个序列玩游戏。

这个游戏的过程非常简单，小 H，小 L 与小 J 分别选取一个 $\{1, 2, \dots, n\}$ 的子集（可以为空，也可以为全集），要求三个人选择的子集**互不相同**，

定义一个子集 $S = \{v_1, v_2, \dots, v_k\}$ 的美丽度为 $a_{v_1} + a_{v_2} + \dots + a_{v_k}$ 。三个人分别选择好子集后，小 H 计算出三个人选择的子集的美丽度，并将三个美丽度相乘。得到的这个值称为一次游戏的价值。

现在小 H 想要知道，对于所有可能的游戏过程，这些游戏的价值之和是多少？由于这个值可能非常大，请输出答案对 1 000 000 007 取模的结果。

输入格式

第一行一个正整数 n 表示序列的长度。

第二行 n 个正整数 a_1, a_2, \dots, a_n 表示序列中的元素。

输出格式

一行一个非负整数表示答案。

样例输入 1

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

样例输出 1

```
12258000
```

数据范围与约定

对于 100% 的数据，满足以下要求：

- $1 \leq n \leq 10^5$ 。
- $1 \leq a_i \leq 10^9$ 。

对于 20% 的数据，满足 $1 \leq n \leq 8$ 。

对于另外 20% 的数据，满足 $1 \leq n \leq 15$ 。

对于另外 20% 的数据，满足 $1 \leq n \leq 1000$ 。