

NOIP 2023 模拟赛

题目名称	最大公因数	博弈	饼干	游戏
题目类型	传统题	传统题	传统题	传统题
目录	divisor	bet	biscuit	game
可执行文件名	divisor	bet	biscuit	game
输入文件名	divisor.in	bet.in	biscuit.in	game.in
输出文件名	divisor.out	bet.out	biscuit.out	game.out
每个测试点时限	1 秒	3 秒	1 秒	2 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	3	4	6	5
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	divisor.cpp	bet.cpp	biscuit.cpp	game.cpp
-----------	-------------	---------	-------------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

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/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. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

最大公因数 (divisor)

【题目描述】

YYX 是个无忧无虑的女孩子。

这天，YYX 在空地上发现了一个长度为 n 的正整数数组，她觉得这个数组太过丑陋了！具体的，我们称一个数组是美丽的，当且仅当这个数组的最大公因数严格大于 1。特别的，我们称空数组也是美丽的。

为了把这个数组改造成美丽的，YYX 决定把若干个元素剔除，但是剔除太多元素就会让数组总和变得过小。所以她找到了你，希望你帮她求出，能否剔除若干个元素后把这个数组变成美丽的，并在此基础上使剩余元素的总和最大，你只需要告诉她这个最大值即可。

【输入格式】

从文件 *divisor.in* 中读入数据。

第一行一个正整数 n 。

第二行一个长度为 n 的正整数序列 a_i ，表示 YYX 发现的数组。

【输出格式】

输出到文件 *divisor.out* 中。

输出一个整数表示保留元素总和最大的美丽的数组的总和。

【样例 1 输入】

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

【样例 1 输出】

```
1 12
```

【样例 1 解释】

其中一个最优方案是保留 $\{2, 4, 6\}$ 。

【样例 2 输入】

```
1 4
2 1 1 1 1
```

【样例 2 输出】

```
1 0
```

【样例 2 解释】

唯一的方案就是保留 {}, 即把所有元素都删了。

【样例 3 输入】

```
1 10
2 28851 8842 9535 2311 25337 26467 12720 10561 8892 6435
```

【样例 3 输出】

```
1 56898
```

【数据范围】

对于全部的数据, $3 \leq n \leq 1000, 1 \leq a_i \leq 10^9$ 。

子任务编号	特殊限制	分值
1	$n \leq 20$	30
2	$a_i \leq 5000$	30
3	无特殊限制	40

博弈 (bet)

【题目描述】

YYX 是个无忧无虑的女孩子。

这天, YYX 在空地上发现了一个长度为 n 的正整数数组, 她觉得这个数组很有意思! 所以她找到了 Dr. Wu, 希望和他在这个数组上做游戏。

具体的, YYX 和 Dr. Wu 事先指定了一个正整数 x , 并在数组上轮流操作, YYX 先手: 当一个玩家操作时, 她将选择数组中一个大于 0 的元素 a_i 并选择一个 $[1, \min(a_i, x)] \cap \mathbb{Z}$ 中的数 j , 令 a_i 减去 j , 不能操作者负。

现在, YYX 找到了你, 希望你对于每个 $1 \leq x \leq n$, 求出若两个人都使用最优策略, 是先手必胜还是后手必胜。

【输入格式】

从文件 *bet.in* 中读入数据。

第一行一个正整数 n 表示数组长度。

第二行一个长度为 n 的正整数序列 a_i , 表示 YYX 发现的数组。

【输出格式】

输出到文件 *bet.out* 中。

输出 n 行, 若 $x = i$ 时先手必胜, 则第 i 行输出一行 First, 否则输出一行 Second。

【样例 1 输入】

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

【样例 1 输出】

```
1 Second
2 First
3 Second
4 First
```

【样例 2 输入】

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

【样例 2 输出】

```
1 Second
2 First
3 Second
4 Second
5 Second
```

【样例 3】

见选手目录下的 *bet/bet3.in* 与 *bet/bet3.ans*。

【数据范围】

对于全部的数据， $1 \leq n \leq 5 \times 10^5, 1 \leq a_i \leq n$ 。

子任务编号	特殊限制	分值
1	$n \leq 8$	10
2	$n \leq 200$	20
3	$n \leq 2000$	30
4	无特殊限制	40

饼干 (biscuit)

【题目描述】

YYX 是个无忧无虑的女孩子。

这天, YYX 在空地上发现了一本魔法书, 书上记载了 n 条魔咒, 其中第 i 条魔咒需要花费 a_i 点蓝。作为一名见习魔法师, YYX 初始拥有 m 点蓝以及 k 块魔法饼干。

YYX 充满好奇心, 她在得到魔法书的那一刻就想要把魔法书里的每一条魔咒都至少使用一次。但是她自己的蓝太少了, 而补蓝需要花钱。具体的, YYX 可以做下列三种操作:

- 选择一条未使用过的魔咒, 假设选择的魔咒是第 i 条。如果 YYX 持有的蓝大于等于 a_i 点, 那么她持有的蓝减去 a_i 点, 并记录第 i 条魔咒为使用过。
- 如果还有魔法饼干, YYX 可以吃掉一个魔法饼干并选择一条魔咒, 假设选择的魔咒是第 i 条。如果此时 $a_i \geq 1$, 那么令 a_i 减一。
- 假如现在 YYX 持有 a 点蓝, 且 $a < m$, 那么 YYX 可以花费 $m - a$ 块钱来使自己持有的蓝加一。也就是说, 假如 $m = 4, a = 1$, 那么 YYX 把蓝补满的花费为 $3 + 2 + 1 = 6$ 。

聪明的你一定知道 YYX 把每条魔咒都使用过一遍所需最少的钱了, 请把这个数目告诉她吧。注意, 你不需要最小化饼干的使用数, 也不需要关心最后 YYX 持有多少蓝。

【输入格式】

从文件 **biscuit.in** 中读入数据。

第一行三个正整数 n, m, k 。分别表示魔咒数量, YYX 初始蓝数, 魔法饼干数量

第二行 n 个整数 a_1, a_2, \dots, a_n , 表示魔咒的花费。

【输出格式】

输出到文件 **biscuit.out** 中。

第一行一个整数表示 YYX 把每条魔咒都使用一遍所需最少钱数。

【样例 1 输入】

```
1 2 4 0
2 2 4
```

【样例 1 输出】

1 3

【样例 1 解释】

其中一种最优操作为：YYX 先使用第一条魔咒，此时 YYX 持有 2 点蓝，之后 YYX 补两次魔，花费 $2 + 1 = 3$ 块钱，最后 YYX 使用第二条魔咒，总花费钱数为 3。

【样例 2 输入】

1 3 16 2
2 6 9 9

【样例 2 输出】

1 21

【样例 2 解释】

YYX 先吃两个饼干，使 a_2, a_3 都变成 8，然后使用第一条魔咒，此时 YYX 持有 10 点蓝，然后 YYX 补六次魔，花费 $6 + 5 + 4 + 3 + 2 + 1 = 21$ 块钱，最后 YYX 依次使用第二、三条魔咒。

【样例 3 输入】

1 3 9 1
2 2 3 9

【样例 3 输出】

1 6

【样例 3 解释】

YYX 吃一个饼干，使 a_2 变为 2，然后使用第一条魔咒，补两次魔，花费 $2 + 1 = 3$ 块钱，使用第二条魔咒，补两次魔，花费 $2 + 1 = 3$ 块钱，最后使用第三条魔咒。总花费为 6，可以证明没有更优的方案。

【样例 4】

见选手目录下的 *biscuit/biscuit4.in* 与 *biscuit/biscuit4.ans*。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 10^5, 1 \leq m \leq 10^6, 1 \leq a_i \leq m, 0 \leq k \leq \sum a_i$ 。

子任务编号	$n \leq$	$m \leq$	特殊限制	分值
1	20	10^6	$k = 0$	10
2	10^5	5000	$k \leq 5000$	10
3	10^5	5000	无	20
4	10^5	10^6	$k = 0$	30
5	10^5	10^6	$\sum a_i \leq 10^6$	10
6	10^5	10^6	无	20

游戏 (game)

【题目描述】

YYX 是个无忧无虑的女孩子。

这天, YYX 在空地上发现了一个长度为 n 的正整数数组, 她觉得这个数组很有意思! 所以她找到了 Dr. Wu, 希望和他在这个数组上做游戏。

具体的, YYX 和 Dr. Wu 在数组上轮流操作, YYX 先手: 当一个玩家操作时, 她 will 选择数组的开头元素或者结尾元素, 然后把它取走, 一个元素被取走了就相当于从数组里删去了。当数组里没有元素时, 两个人比较自己取走的所有元素的 按位异或和, 大的一方获胜, 若一样则平局。

然而 YYX 发现自己总是赢不了 Dr. Wu, 于是她找到了你, 希望你求出, 在两人都采用最优决策的情况下, 是先手必赢还是后手必赢还是平局。

【输入格式】

从文件 *game.in* 中读入数据。

本题采用多组数据测试。

第一行一个正整数 T 表述数据组数。

接下来, 对于每组数据, 第一行一个正整数 n 表示数组长度, 第二行一个长度为 n 的正整数序列 a_i , 表示 YYX 发现的数组。

对于每个测试点, 所有数据的 n 之和不超过 10^5 。

【输出格式】

输出到文件 *game.out* 中。

对于每组数据, 假如先手必胜, 输出一行 First; 假如后手必胜, 输出一行 Second; 假如平局, 输出一行 Draw。

【样例 1 输入】

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

```
7 29 24 15 12 30 13 19
8 6
9 6 2 5 6 3 4
```

【样例 1 输出】

```
1 First
2 Second
3 First
4 Draw
```

【样例 2】

见选手目录下的 *game/game2.in* 与 *game/game2.ans*。

【数据范围】

对于全部的数据， $\sum n \leq 5 \times 10^5, 1 \leq n \leq 10^5, 0 \leq a_i < 2^{30}$ 。

子任务编号	特殊限制	分值
1	$n \leq 200, a_i \in \{0, 1\}$	5
2	$n \leq 200$	15
3	$n \equiv 0 \pmod 2$	5
4	$\forall 1 \leq i \leq \lfloor \frac{n}{2} \rfloor, a_{2i-1} = a_{2i}$	15
4	无特殊限制	60