2022 年湖北省信息学能力线上测试

入门组

正式赛

时间：2022 年 11 月 27 日 08:30 ～ 12:00

题目名称	语言学习	魔法少女	01 游戏	猜数博弈
题目类型	传统型	传统型	传统型	传统型
目录	`study`	`magic`	`game`	`gamble`
可执行文件名	`study`	`magic`	`game`	`gamble`
输入文件名	`study.in`	`magic.in`	`game.in`	`gamble.in`
输出文件名	`study.out`	`magic.out`	`game.out`	`gamble.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++　语言	`study.cpp`	`magic.cpp`	`game.cpp`	`gamble.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内存限制一致。
8. 统一评测时采用的机器配置为 Inter(R) Core(TM) i5-10400 CPU @ 2.90GHz，内存 16GB。上述时限以此配置为准。
9. 只提供 Linux 格式附加样例文件。
10. 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。

语言学习（study）

【题目描述】

某 E 最近正在学习 Python 语言，她认为 Python 语言提供的一些字符串处理函数非常便利。

某 E 尤其钟爱 `title()` 函数与 `strip()` 函数。

对于一个由若干个英文单词和空格组成的字符串，`title()` 函数可以将该字符串按**单词首字母大写，其他字母小写**的方式显示。

例如，输入字符串 `hEllo expecT two Thousand AND four`，下面的 Python 程序会输出 `Hello Expect Two Thousand And Four`。

```
s = input()
t = s.title()
print(t)
```

`strip()` 函数可以去除字符串左右两侧的空格。

例如，输入字符串 `abc abc`（两侧各有一个空格），下面的 Python 程序会输出 `abc abc`（两侧均无空格）。

```
s = input()
t = s.strip()
print(t)
```

现在，某 E 希望你使用指定的竞赛语言编写程序，读取一个由英文单词和空格组成的字符串 S，输出依次对其执行 `title()` 函数和 `strip()` 函数的结果。

换句话说，请你使用指定的竞赛语言，实现下面 Python 程序的功能。

```
s = input()
t1 = s.title()
t2 = t1.strip()
print(t2)
```

【输入格式】

从文件 ***study.in*** 中读入数据。

输入一行一个仅由英文字母和空格组成的字符串 S。

【输出格式】

输出到文件 ***study.out*** 中。

输出一行一个字符串，为按题要求处理后的字符串。

【样例 1 输入】

```
title
```

【样例 1 输出】

```
Title
```

【样例 2 输入】

```
  hUbei Province
```

【样例 2 输出】

```
Hubei Province
```

【样例 3 输入】

```
 let  US dO iT  BRAVELY
```

【样例 3 输出】

```
Let  Us Do It  Bravely
```

【样例 4】

见选手目录下的 ***study/study4.in*** 与 ***study/study4.ans***。

【子任务】

对于 30% 的测试数据，S 中不含有空格；
对于另外 30% 的测试数据，S 仅在左右两侧含有空格；
对于 100% 的测试数据，$1 \leq |S| \leq 10^6$，S 仅由大小写英文字母和空格组成。

魔法少女（magic）

【题目描述】

某 E 是一名魔法少女，她可以对数字实施神奇的魔法。

一天，某 E 接到了一个任务。她需要施展魔法，将数字 a 变成数字 b。

某 E 经过考察，决定只使用以下三种魔法，每一种魔法均可施展任意次：

1. 删去数字 a 的个位，即令 $a = \lfloor \frac{a}{10} \rfloor$；
2. 在数字 a 的末尾增加一个 0，即令 $a = a \times 10$；
3. 将数字 a 的个位修改为另一个数。

每施展一次魔法，均需消耗 K 单位灵力。某 E 想要知道，完成该任务，她最少消耗多少单位灵力。

【输入格式】

从文件 ***magic.in*** 中读入数据。

输入共三行，依次为三个正整数 a, b, K。

【输出格式】

输出到文件 ***magic.out*** 中。

输出一个行一个整数，代表某 E 最少消耗多少单位灵力。

【样例 1 输入】

```
12365
12345
2
```

【样例 1 输出】

```
8
```

【样例 2 输入】

```
20222
20222
6
```

【样例 2 输出】

```
0
```

【样例 3】

见选手目录下的 ***magic/magic3.in*** 与 ***magic/magic3.ans***。

【子任务】

对于 20% 的测试数据，$1 \le a, b \le 10$；
对于 70% 的测试数据，$1 \le a, b \le 10^9$；
对于 100% 的测试数据，$1 \le a, b \le 10^{1000000}, 1 \le K \le 10^6$，保证 a, b 位数相同。

【提示】

$\lfloor x \rfloor$ 代表 x 向下取整。例如，$\lfloor 1.2 \rfloor = 1, \lfloor 4 \rfloor = 4, \lfloor -2.8 \rfloor = -3$。

01 游戏（game）

【题目描述】

某 E 喜欢玩 01 游戏。一次 01 游戏会涉及到一个长度为 n 的数字环，环上的每一个位置是数字 0 或 1。环上每个数字按顺序依次编号为 $1, 2, 3, \cdots, n$，x 与 $x-1, x+1$ 相邻。特别的，1 与 $2, n$ 相邻，n 与 $1, n-1$ 相邻。

在每一轮游戏中，某 E 可以选择一个数字为 1 的位置 x，并将 x 与其相邻的两个位置上的数删去。剩余的数字仍构成一个环。

例如，下面的图片展示了红框选中的数字删除前后数字环的变化。

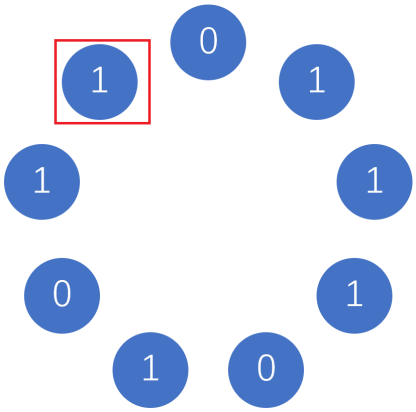

图 1: 删除前

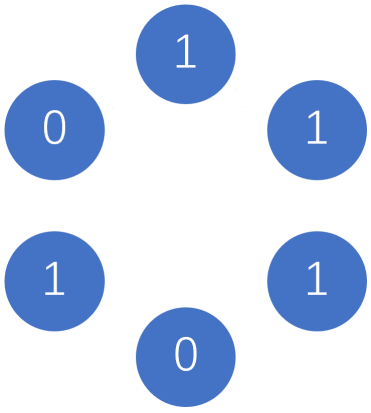

图 2: 删除后

某 E 一共玩了 Q 次 01 游戏。某 E 想要知道，对于每一次游戏，是否存在一种方案，使得在若干轮游戏后，整个数字环被全部删除。

【输入格式】

从文件 ***game.in*** 中读入数据。

输入的第一行为一个整数 Q，代表游戏的次数。

接下来 Q 行，每行输入 $n+1$ 个数。第一个正整数为 n，代表数字环的长度。接下来 n 个正整数，第 i 个代表数字环上位置 i 对应的数。

【输出格式】

输出到文件 ***game.out*** 中。

输出 Q 行，每行一个字符串 `Yes` 或 `No`，分别代表能否完成。

【样例 1 输入】

```
2
6 0 0 0 0 0 0
6 1 1 1 1 1 1
```

【样例 1 输出】

```
No
Yes
```

【样例 2 输入】

```
2
9 1 0 1 0 1 0 1 0 1
9 1 1 1 0 0 1 1 1 1
```

【样例 2 输出】

```
Yes
Yes
```

【样例 3 输入】

```
2
9 1 1 1 1 0 0 0 0 0
9 1 1 1 1 0 0 0 1 0
```

【样例 3 输出】

```
No
Yes
```

【样例 4】

见选手目录下的 ***game/game4.in*** 与 ***game/game4.ans***。

【子任务】

对于 20% 的数据，$n = 3$；
对于另外 30% 的数据，$n \leq 12$；
对于另外 20% 的数据，连续 1 的个数不超过 2；
对于 100% 的数据，$1 \leq n \leq 10^5, 1 \leq Q \leq 20$，保证 n 是 3 的倍数。

猜数博弈（gamble）

【题目描述】

某 E 和 L 队在玩游戏，两个人都绝顶聪明。

游戏开始的时候，某 E 和 L 队额头上分别有一个正整数（某 E 和 L 队知道自己和对方头上都是正整数）。但是，他们都只能看到对方额头上的数字，而不知道自己额头上的数字是什么。

假定某 E 头上数字是 a，L 队头上数字是 b，根据游戏规则，这两个数一定满足 $a = 2b$ 或 $b = 2a$（某 E 和 L 队也知道这一点）。

由某 E 开始，两人轮流进行猜数。如果当前猜数的人能够肯定自己头上的数字是什么，她会直接说出这个数，并获得胜利；否则她会说" 我不知道"，之后由另一个人进行猜数。

现在，告诉你某 E 和 L 队头上的数字 a, b，如果你认为两人都无法在有限次猜数中确知自己头上的数字，输出 `No,Commander`，否则你需要回答以下问题中的一个或者两个：

问题 1：需要经过多少次猜数游戏才会结束；

问题 2：获胜者的名字（`E` 或者 `L`）。

【输入格式】

从文件 ***gamble.in*** 中读入数据。

每个测试点中包含多组测试数据。

输入的第一行为两个正整数 T, op，表明该测试点中测试数据数目和你需要回答的问题。

- 若 $op = 1$，表明你只需要回答问题 1；
- 若 $op = 2$，表明你只需要回答问题 2；
- 若 $op = 3$，表明你需要同时回答两个问题。

接下来 T 行，每行两个正整数 a, b，表明这一组数据中某 E 和 L 队头上的数字分别是什么。

【输出格式】

输出到文件 ***gamble.out*** 中。

输出 T 行，如果在本组测试数据中，两人都不能在有限次猜数中确知自己头上的数字，输出一行一个字符串 `No,Commander`。

否则：

- 若 $op = 1$，输出一行一个正整数，表示问题 1 的答案；
- 若 $op = 2$，输出一行一个字符串，表示问题 2 的答案；

- 若 $op = 3$，输出一行，包括一个正整数和一个字符串，二者间用一个空格隔开，表示问题 1 和问题 2 的答案。

【样例 1 输入】

```
5 1
1 2
2 1
12 24
3 6
9 18
```

【样例 1 输出】

```
2
1
4
2
2
```

【样例 2】

见选手目录下的 ***gamble/gamble2.in*** 与 ***gamble/gamble2.ans***。

【子任务】

对于 25% 的测试数据，$op = 1$；

对于另外 25% 的测试数据，$op = 2$；

对于另外 15% 的测试数据，$op = 3, a = 2^k$，k 为整数。

对于 100% 的测试数据，$1 \le T \le 2 \times 10^5, 1 \le a, b \le 10^{18}$，保证 $a = 2b$ 或 $b = 2a$，$op \in \{1, 2, 3\}$。

请注意，子任务仅限制输入数据的大小。某 E 和 L 队并不知道对 a, b 的范围限制，两人仅知道 a, b 是正整数。