

A 破译Yt

题目描述

笨蛋 Yt 使用了并不聪明的加密方式来传递消息。在他加密这个仅由小写字母组成的消息时，他将每个编号（a为0，b为1，依次加一，以此类推）与其位置（下标从0开始）对 26 取余后相等的小写字母替换成了随机的大写字母。现在给你一个加密后的字符串，你能帮助 Red 破译吗？

输入格式

一行，一个字符串，长度不超过 100。

输出格式

一行，一个字符串，表示解密后的字符串，全部由小写字母组成。

样例输入

```
QwQ
```

样例输出

```
awc
```

样例解释

awc 中第一个字母 a 编号为0，与位置0相同，故被替换为一个任意的大写字母；第二个字母 w 编号为22，与位置不同，不被替换； c 编号为2，与位置2相同，故也被替换为一个大写字母

Aurthor : Red

B 分久必合 合久必分

题目描述

现有若干行包含**大小写字母与空格**的字符串和操作标识。

- 若字符串前操作标识为 0，将包含空格的字符串整体输出；
- 若字符串前操作标识为 1，将字符串按空格拆开分行输出。

输入

第一个数为字符串行数 n 。

接下来 n 行，每行组成为 [操作标识][空格][包含空格的操作字符串]

输出

输出若干行，即按照操作标识的输出结果

输入样例

```
3
0 str1 str2 str3
1 str4 str5 str6
0 str7
```

输出样例

```
str1 str2 str3
str4
str5
str6
str7
```

数据范围

- $n \leq 100$
- 操作字符串长度 < 100
- 字符串不会以空格开头，结尾，也不会出现两个连续空格

Author: Olivaw

C 收集石樱

题目描述

熙熙攘攘的旧地狱街道，不时飘落点点樱花，那是灵魂的结晶所凝结而成的粉色石头，被称为「石樱」。

石樱是怨灵们的最爱。

有一个怨灵正在旧地狱收集石樱。石樱分布在编号从 1 到 n 的 n 个节点上。起初，第 i 个节点上的石樱数量为 a_i 。

它可以选择一个节点作为出发地，每经过一个单位时间，会依次发生如下事件：

1. 怨灵从节点 a 飘到 b ; ($|a - b| \leq 1$)
2. 它收集节点 b 的所有石樱;
3. 每个节点长出一个新的石樱。

它想知道，在 m 个单位时间内，最多能收集多少石樱。

输入

第一行一个整数 T ，表示数据组数。

对于每组数据：

第一行两个整数 n, m ，含义见上；

第二行 n 个整数 a_1, a_2, \dots, a_n ， a_i 表示第 i 个节点的初始石樱数量。

输出

每组数据一行一个整数，表示 m 个单位时间内收集的石樱数量最大值。

样例

输入

```
2
8 6
2 6 9 5 10 3 8 7
8 11
3 2 2 2 7 2 1 7
```

输出

```
57
78
```

数据范围

$1 \leq T \leq 10^4$, $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$, $1 \leq m \leq 10^9$ 。

对于每个测试点， $\sum n \leq 2 \times 10^5$ 。

D 阿求的岁月史书PLUS

题目描述

在**永夜异变**时期，为了保护人类的村落，上白泽慧音将其从历史上抹除，使任何人都无法发现村落的存在。

但稗田家代代相传的《幻想乡缘起》属于她的能力触及范围之外，因此幻想乡缘起中仍然存在有关村落的记录，请你从《幻想乡缘起》中找到和村落有关的内容，将其替换为一段等长的伪造的历史。

输入格式

输入共 $3 \times T + 1$ 行

第一行为数据组数 T

接下来为 T 组数据，每组数据共三行，其中：

- 第一行为待修改的《幻想乡缘起》的片段，字符串 S
- 第二行为有关人类村落的历史片段，字符串 s
- 第三行为一段伪造的历史，字符串 s'

字符串均为仅由大小写字母组成的非空串

保证 s 和 s' 的长度相等

输出格式

输出共 T 行

对于每组数据，输出一行：

- 若 S 中存在 s ，则将**首次**出现的 s 替换为 s' ，然后输出替换后的 S
- 若 S 中不存在 s ，则输出一行字符串 **I'll make this night never have happened!**

样例

输入样例

```
2
KamishirasawaKeine
Kami
Were
ImperishableNight
TrueMoon
FakeMoon
```

输出样例

```
wereshirasawaKeine
I'll make this night never have happened!
```

样例解释

共两组数据

第一组数据，字符串 `KamishirasawaKeine` 中存在 `Kami`，将其替换为 `Were`，输出替换后的字符串 `WereshirasawaKeine`

第二组数据，字符串 `ImperishableNight` 中不存在 `TrueMoon`，输出 `I'll make this night never have happened!`

数据范围

$1 \leq T \leq 10$

S, s, s' 均不为空串且长度不超过 50

Hint

`string.h` 库中的函数 `strstr` 可以查找字符串 S 中是否存在字符串 s ，若存在则返回其中包含的 s 所在的位置的指针。

`string.h` 库中的另一个函数 `memcpy` 可以将一个数组中复制若干个元素进入另一个数组中，使用例如下

```
int a[5] = {0, 1, 2, 3, 4}, b[5] = {4, 3, 2, 1, 0};
memcpy(a, b, 3 * sizeof(int));
// 操作之后 a 数组的元素变为 {4, 3, 2, 3, 4}
```

或者，直接用遍历替换甚至不进行替换也是可行的

E 坤坤打篮球

题目描述

众所周知，坤坤的篮球水平很强，不说是能随意过奥尼尔，那也能跟库里打得有来有回。在篮球比赛中，罚球得分为1分，上篮和三分线内进球得2分，三分线外进球得3分。现在已知坤坤预计在KBA(kunkun的专属篮球赛)中得了 n 分，作为一个合格的ikun，请你帮坤坤算算他有多少种得分情况可以得到 n 分。

输入格式

一行正整数 t ,表示有 t 组输入。

接下来 t 行，每行一个正整数 $n(1 \leq n \leq 20)$ 。

输出格式

t 个正整数。

输入样例

```
1
3
```

输出样例

```
4
```

样例解释

得分情况分别为：

1. 1+1+1
2. 1+2
3. 2+1
4. 3

author: HiDen



F 这就是爵士乐吗？

题目背景：

来到日本涩谷区的孔明先生正在学习流行音乐。他在钢琴上弹了四个音，你要确定这四个音组成的和弦是否来自一条音阶。

钢琴有很多个八度，一个八度之间的十二个半音由低到高依次为 **C, C#, D, D#, E, F, F#, G, G#, A, A#, B**。**#** 代表升高一个半音，**b** 代表降低一个半音。十二个半音首尾相接围成一圈，**B#** 就是 **C**，而 **Cb** 就是 **B**。由于升降号可以叠加，同一个音高可以有很多种表示。比如 **B#####**，**C#####**，**D####**，**E#**，**F**，**Gbb** 表示的是同一个音 **F**。

虽然将升降号写在字母前后的两种做法都存在，但本题的输入中，我们统一将其写在后面。

一个自然调式有七个音。比如 Ionian scale 的七个音是 **C D E F G A B**。如果同时升降 x 个半音后，和弦的所有组成音都与 **C D E F G A B** 相同（可以重复），那么认为这个和弦来自一条音阶。

说明：

通过与 **C** 的半音高差，可以确定每个音的音高。**CDEFGAB** 相对于 **C** 的半音高差分别为 0, +2, +4, +5, +7, +9, +11。每升半音，与 **C** 的半音高差升高 1，降半音则半音高差减少 1。半音高相差 12 的整数倍的音是同名的（相同的）。

注意解决负数取模的问题。

输入：

第一行，一个整数，表示测试数据组数。

随后每行四个由 `ABCDEFG#b` 组成的字符串，各表示一个音。`#` 和 `b` 不会都出现在同一个字符串中。

输出：

每行输出一个字符串。如果这个和弦来自一条音阶，输出 `harmonic`。否则输出 `SOUNDS JAZZ`。

样例输入：

```
3
A B C Db
D Gb A A###
F F F F
```

样例输出：

```
SOUNDS JAZZ
harmonic
harmonic
```

##解释和提示：

$G_b = F\#$ ， $A\#\#\# = B\# = C$ 。

样例第二个和弦等价于Lydian Mode的二级和弦，D属七和弦（ $D F\# A C$ ）。将每个音升高五个半音可以得到G属七和弦（ $G B D F$ ），这是一个Ionian scale调内的和弦。

从半音高的角度，这四个音分别是2，6，-3，0，同时加5得到7，11，2，5。

另外，使用 `scanf("%s",&string)` 和 `strlen(string)` 可以帮你快速解题。

数据范围

输入字符数之和不超过 20000。

author: Yt

G 表达式的值

题目背景

void有一个表达式，只由正整数、加号和乘号构成

void想让你求出这个表达式的值

保证所有输入的正整数都在 int 范围内

保证计算的结果在 $longlong$ 范围内

输入

输入一行，不超过200个字符，中间没有空格

输出

输出一个正整数，表示表达式的值

样例

样例输入

```
1*2*3+5+7+2*4
```

样例输出

```
26
```

数据范围

输入只由正整数、加号和乘号构成

计算的结果在 $longlong$ 范围内

H 打音游！！！

题目背景

bluebean 上次打音游打得并不过瘾。他觉得音游应该容纳更多元素！于是在这次的打音游过程中，他记下了自己的**连击**与收集**音符**、**红心**的情况，想据此直接计算出自己的分数。你能帮帮他吗？

题目描述

输入情况中存在 P p G g M m n H 共六种情况。敌人分为**大型敌人**与**小型敌人**，对敌人的击打情况分为 *Perfect*、*Great*、*Miss*。除击打敌人外还能够收集到**音符**与**红心**。下面的表格对各个输入的情况做出了解释。

输入	含义
P	成功击打 大型敌人 并取得 <i>Perfect</i>
p	成功击打 小型敌人 并取得 <i>Perfect</i>
G	成功击打 大型敌人 并取得 <i>Great</i>
g	成功击打 小型敌人 并取得 <i>Great</i>
M	未能成功击打 大型敌人 ，取得 <i>Miss</i>
m	未能成功击打 小型敌人 ，取得 <i>Miss</i>
n	收集到 音符
H	收集到 红心

每次成功的击打敌人（即 *Perfect* 与 *Great*）将会使 *Combo*（**连击数**）加一。如果未能成功击打敌人（即 *Miss*），则 *Combo* 数归零。游戏开始时 *Combo* 数为 0。

击打敌人得到的分值计算方法：

对于每次击打，收获的分值 = 基础分值 x 判定系数 x 连击倍率

基础分值与击打敌人的种类有关：击打大型怪物的基础分值为 300，击打小型怪物的基础分值为 200

判定系数与击打情况相关：

Perfect : 1.0

Great : 0.6

Miss : 0

连击倍率与 *Combo* 数有关：

1~9 *Combo*: 1.0

10~19 *Combo*: 1.1

20~29 *Combo*: 1.2

30~39 *Combo*: 1.3

40~49 *Combo*: 1.4

50 *Combo* 及以上: 1.5

注：在击打成功时会先将 *Combo* 数加一再计算对应的倍率。

收集**音符**不会增加 *Combo* 数，但是会增加分数。每收集一个音符增加的分数：150

在游戏中存在**生命值系统**：**初始生命值**及**生命值上限**都为 200，每次取得 *Miss* 都会损失一定生命值。未能成功击打大型敌人（即 **M**）会损失 40 点生命值，未能成功击打小型敌人（即 **m**）会损失 20 点生命值。

每收集一个**红心**（即 **H**）会恢复 50 点生命值。生命值不会超过血量上限。特别地，如果收集到红心时处于满血状态，会额外增加 600 点分数。收集红心不会增加 *Combo* 数。

最终得到的分数为游戏过程中击打敌人、收集音符和红心所得到的所有分数的加和。特别地，如果在游戏过程中生命值小于等于零，则判定为死亡，**不会统计**死亡后获得的分数（即生命值归零后的输入都不会计入分数）。

本题存在多组输入数据。

输出时，对于每组数据：

先输出一行一个数字，代表最终得分。

此外：

若对敌人的击打情况没有出现 *Great* 和 *Miss*，额外输出一行 **All Perfect!**。

若对敌人的击打情况没有出现 *Miss*，额外输出一行 **Full Combo!**（若已经输出了 **All Perfect!** 则不再输出 **Full Combo!**）。

若在游戏过程中死亡，额外输出一行 **Game Over!**。

输入格式

不定行输入，每行一个字符串代表游戏情况，保证字符串中的字符均为 `P p G g M m H n` 中的一种，且字符之间不存在空格。

输出格式

对于每行输入，先输出一个整数代表最终得分。

若对敌人的击打情况没有出现 *Great* 和 *Miss*，额外输出一行 `All Perfect!`。

若对敌人的击打情况没有出现 *Miss*，额外输出一行 `Full Combo!`（若已经输出了 `All Perfect!` 则不再输出 `Full Combo!`）。

若在游戏过程中死亡，额外输出一行 `Game Over!`。

输入样例1

```
PPPMGGG
```

输出样例1

```
1440
```

输入样例2

```
pnGpnPnnpnnnnnGPPnppngnppnpnnPn
```

输出样例2

```
5652  
Full Combo!
```

数据范围

对于 30% 的数据，输入的字符串长度不超过 30

对于 100% 的数据，输入的字符串长度不超过 10000

对于所有数据，输入行数不超过 10 行。

Hint

本题是一道大模拟，考察你的综合代码能力。在开始码代码前要仔细思考好代码的结构喔。

如果你想拿上次的代码改的话，需要注意判定系数改变了喔。

优雅地使用 *for* 和 *if* 解决本题吧~

Author : bluebean

I 字符串大练习

题目描述

? 给出一个待操作字符串 *str*，请实现对其的增删改查。具体要求如下：

输入

? 第一行一个字符串 *str*

? 接下来不定行，每行表示一次操作，可能为以下情况之一，每行不同内容之间均用一个空格隔开，以下出现的 *i, j* 表示的位数均**从 0 开始计**：

1. 尾接：一个数字 **1**，一个字符串 *s*，表示将 *s* 接到 *str* **末尾**；
2. 尾删：一个数字 **2**，一个整数 *i*，表示删除 *str* 的**第 *i* 位及其之后**的字符；
3. 插入：一个数字 **3**，一个整数 *i*，一个字符串 *s*，表示将 *s* 插入到 *str* 的第 *i* 位**之前**，后面的字符**向后顺移**；
4. 删除：一个数字 **4**，两个整数 *i, j* ($i \leq j$)，表示删除 *str* 的第 *i* 到 *j* 个字符（**包括第 *i* 和第 *j* 个**），后面的字符**向前补位**；
5. 修改：一个数字 **5**，一个整数 *i*，一个字符串 *s*，表示从 *str* 的第 *i* 位开始用 *s* 覆盖，覆盖是指，从第 *i* 位开始，将 *str* 的字符依次替换为 *s* 的字符，如果 *str* 长度不够，则**向后延长直到 *s* 结束**，如果 *s* 长度不到 *str* 结尾，则后续字符**保留 *str* 原字符**；
6. 结束：一个数字 **-1**，表示结束操作。

? 最后需要查询字符串出现次数：一个字符串 *s*，请**输出 *s* 在 *str* 中出现的次数**（**重叠出现也计算在内**）。

输出

? 输出共两行

? 第一行一个整数表示查询结果，即最后查询的字符串 *s* 在 *str* 中出现的次数（**重叠出现也计算在内**）

? 第二行一个字符串表示经过若干次操作后最后的 *str*

样例

输入

```
aaaaa
1 bbbbbbb
2 10
3 2 bbb
4 7 9
5 8 aabaa
5 1 baa
-1
abaa
```

输出

```
4
abaabaabaabaa
```

样例解释

每次操作后 *str* 如下：

```
第1次操作 -1尾接：aaaaabbbbbbb
第2次操作 -2尾删：aaaaabbbbb
第3次操作 -3插入：aabbbaaabbbbb
第4次操作 -4删除：aabbbbaabbb
第5次操作 -5修改：aabbbbaabaabaa
第6次操作 -5修改：abaabaabaabaa
```

最后一次操作之后 *str* 中有 4 个 **abaa**

数据范围

str, *s* 均仅由小写字母组成

保证操作过程中 *str*, *s* 长度不超过 10000

操作次数不超过 1000

$0 \leq i \leq j < len$, *len* 为 *str* 长度

Hint

? 可能用到头文件 *string.h* 中函数 *strlen*, *strcat*, *strcpy*, *memcpy*, *strstr*

Author: 哪吒

J Yt那山峦般沉重的歉意

题目背景

种种回忆，会留在人们心中。

Yt出难题闹人，已经为这个世界带来了许多痛苦的回忆。幡然醒悟的Yt决定写一封长长的道歉信来表达自己的忏悔。

反思的过程中，Yt对自己的忏悔有了更明确的认识。当把所有的忏悔内容分成 n 项，第 i 项忏悔的明确程度是 $((n + i)^2 \bmod p) - p/2$ 。Yt接下来要挑出所有忏悔中最明确的连续区间，将它转写成充满歉意的告白书。

题目描述

忏悔的明确程度是一个确定的，长度为 n 数列，这个数列的第 i 项可以用 $((n + i)^2 \bmod p) - p/2$ 求出。 n ， p 是给定的不小于 7，不大于 10^7 的正整数， $p/2$ 取向 下取整后的值。

你要选择一个 $[1, n]$ 内的非空区间 $[l, r]$ ，使得数列第 l 到第 r 项的和取得最大值 M 。

输入

一组输入，两个整数，是 n ， p ，含义见题目描述。

输出

输出空格分开的三个整数，表示输入对应的 l ， r ， M 。有多组同时可以取得最大值的时候，输出 l 最小，其次 r 最小的一组。

样例输入

20 37

样例输出

4 12 35

样例解释

20，37 对应的数列为

16 -15 -7 3 15 -8 8 -11 9 -6 18 7 -2 -9 -14 -17 -18 -17 -14 -9

数据范围

$7 \leq n, p \leq 10^7$

花絮

我啊，改悔罢！在这充满了悲伤和痛苦的程设上机，我的罪过罄竹难书！只有终结了充满了优越和偏见的出题人，才能创造美好的新世界！最后的警钟已经敲响，偿还错误的时间终将到来。我 Yt 只知道 C 语言，不知道人性，已经踏上偏离正轨的道路，只知道用难题拷打，而失去了用讲解来阐释的善良...（未完待续）

author: Yt

K 高高地飞起来吧，北京五号！

题目背景

“真正相信奇迹的人，本身就和奇迹一样了不起。” —— 《星游记》

文传源先生是北航自动化科学与电气工程学院及控制学科创建人之一、航空航天控制与系统仿真著名专家。61年前，正是他作为总设计师，带领三百北航师生大干三百天，研制出中国第一架无人驾驶飞机，创造了“北航奇迹”。为了纪念文传源先生，2020年传源书院正式成立。

穿越时空，你来到了北京五号的控制车间，你了解到北京五号可以处理四种指令：

指令编号	指令参数	指令效果
1	d	向正前方前进 d 千米
2	d	向正后方前进 d 千米（不改变原来方向）
3	a	顺时针转向 a 度

指令编号	指令参数	指令效果
4	a	逆时针转向 a 度

现在需要设计一个应急决策系统：当一次性接收 n 个指令时，**调整执行指令的顺序**，使得飞机能够飞的**尽量远**（即最终位置离出发点的水平距离尽量远）。相信这肯定难不倒来自未来的你吧！

输入格式

$n + 1$ 行输入。

第一行一个整数 n ，表示有 n 条指令。

接下来 n 行，每行两个整数 op 和 x ，表示指令编号和指令参数，空格隔开。

输出格式

两行输出。

第一行根据情况输出：

- 如果飞行距离个位是奇数输出：SHIE
- 如果飞行距离个位是偶数输出：CHUANYUAN

第二行一个浮点数（保留六位小数），表示能飞的最远距离。

输入样例

```
4
1 180
3 360
2 180
4 180
```

输出样例

```
CHUANYUAN
360.000000
```

数据范围

n, d, a 均为正整数

$$1 \leq n \leq 10^4$$

$$1 \leq d \leq 10^4$$

$$1 \leq a \leq 10^3$$

赶快做完这题，和Uanu一起去看彩虹海吧！

auhtor: Uanu