**NOIP 提高组 T1 真题**

**NOIP2008 笨小猴(word.pas/c/cpp)**

【问题描述】 笨小猴的词汇量很小，所以每次做英语选择题的时候都很头疼。但是他找到了一种方法，经试验证明，用这种方法去选择选项的时候选对的几率非常大！

这种方法的具体描述如下：假设maxn是单词中出现次数最多的字母的出现次数，minn是单词中出现次数最少的字母的出现次数，如果maxn-minn是一个质数，那么笨小猴就认为这是个Lucky Word，这样的单词很可能就是正确的答案。

【输入】

输入文件word.in只有一行，是一个单词，其中只可能出现小写字母，并且长度小于100。

【输出】

输出文件word.out共两行，第一行是一个字符串，假设输入的的单词是Lucky Word，那么输出“Lucky Word”，否则输出“No Answer”；

第二行是一个整数，如果输入单词是Lucky Word，输出maxn-minn的值，否则输出0。

【输入输出样例1】

|  |  |
| --- | --- |
| word.in | word.out |
| error | Lucky Word  2 |

【输入输出样例1解释】

单词error中出现最多的字母r出现了3次，出现次数最少的字母出现了1次，3-1=2，2是质数。

【输入输出样例2】

|  |  |
| --- | --- |
| word.in | word.out |
| olympic | No Answer  0 |

【输入输出样例2解释】

单词olympic中出现最多的字母i出现了2次，出现次数最少的字母出现了1次，2-1=1，1不是质数。

**NOIP2009 潜伏者（spy.pas/c/cpp）**

【问题描述】R国和S国正陷入战火之中，双方都互派间谍，潜入对方内部，伺机行动。历经艰险后，潜伏于S国的R国间谍小C终于摸清了S国军用密码的编码规则：

1）S国军方内部欲发送的原信息经过加密后在网络上发送，原信息的内容与加密后所的内容均由大写字母‘A’—‘Z’构成（无空格等其他字母）。

2）S国对于每个字母规定了对应的“密字”。加密的过程就是将原信息中的所有字母替换为其对应的“密字”。

3）每个字母只对应一个唯一的“密字”，不同的字母对应不同的“密字”。“密字”可以和原字母相同。

例如，若规定‘A’的密字为‘A’，‘B’的密字为‘C’（其他字母及密字略），则原信息“ABA”被加密为“ACA”。

现在，小C通过内线掌握了S国网络上发送的一条加密信息及其对应的原信息。小C希望能通过这条信息，破译S国的军用密码。小C的破译过程是这样的：扫描原信息，对于原信息中的字母x（代表任一大写字母），找督其在加密信息中的对应大写字母y，并认为在密码里y是x的密字。如此进行下去直到停止于如下的某个状态：

1）所有信息扫描完毕，‘A’—‘Z’所有26个字母在原信息中均出现过并获得了相应的“密字”。

2）所有信息扫描完毕，但发现存在某个（或某些）字母在原信息中没有出现。

3）扫描中发现掌握的信息里有明显的自相矛盾或错误（违反S过密码的编码规则）。例如某条信息“XYZ”被翻译为“ABA”就违反了“不同字母对应不同密字”的规则。

在小C忙得头昏脑胀之际，R国司令部又发来电报，要求他翻译另外一条从S国刚刚截取到的加密信息。现在请你帮助小C：通过内线掌握的信息，尝试破译密码。然后利用破译的密码，翻译电报中的加密信息。

【输入】

输入文件名为spy.in，共3行，每行为一个长度在1到100之间的字符串。第1行为小C掌握的一条加密信息。第2行为第1行的加密信息所对应的原信息。第3行为R国司令部要求小C翻译的加密信息。

输入数据保证所有字符串仅由大写字母‘A’—‘Z’构成，且第1行长度与第2行相等。

【输出】

输出文件spy.out共1行。若破译密码停止时出现2，3两种情况，请你输出“Failed”（不含引号，注意首字母大写，其它小写）。否则请输出利用密码翻译电报中加密信息后得到的原信息。

【输入输出样例1】

|  |  |
| --- | --- |
| Spy.in | Spy.out |
| AA  AB  EOWIE | Failed |

【输入输出样例说明】

原信息中的字母“A”禾B”对应相同的密字，输出“Failed”。

【输入输出样例2】

|  |  |
| --- | --- |
| Spy.in | Spy.out |
| QWERTYUIOPLKJHGFDSAZXCVBN  ABCDEFGHIJKLMNOPQRSTUVWXY  DSLIEWO | Failed |

【输入输出样例2说明】

字母“Z”在原信息中没有出现，输出“Failed”。

【输入输出样例3】

|  |  |
| --- | --- |
| Spy.in | Spy.out |
| MSRTZCJKPFLQYVAWBINXUEDGHOOILSMIJFRCOPPQCEUNYDUMPP  YIZSDWAHLNOVFUCERKJXQMGTBPPKOIYKANZWPLLVWMQJFGQYLL  FLSO | NOIP |

**NOIP2010 机器翻译(translate.pas/c/cpp)**

【问题描述】 小晨的电脑上安装了一个机器翻译软件，他经常用这个软件来翻译英语文章。 这个翻译软件的原理很简单，它只是从头到尾，依次将每个英文单词用对应的中文含义来替换。对于每个英文单词，软件会先在内存中查找这个单词的中文含义，如果内存中有，软件就会用它进行翻译；如果内存中没有，软件就会在外存中的词典内查找，查出单词的中文含义然后翻译，并将这个单词和译义放入内存，以备后续的查找和翻译。 假设内存中有 M 个单元，每单元能存放一个单词和译义。每当软件将一个新单词存入内存前，如果当前内存中已存入的单词数不超过 M?1，软件会将新单词存入一个未使用的内存单元；若内存中已存入 M 个单词，软件会清空最早进入内存的那个单词，腾出单元来，存放新单词。

假设一篇英语文章的长度为 N个单词。给定这篇待译文章，翻译软件需要去外存查找多少次词典？假设在翻译开始前，内存中没有任何单词。

【输入】

输入文件名为 translate.in，输入文件共 2 行。每行中两个数之间用一个空格隔开。

第一行为两个正整数 M和 N，代表内存容量和文章的长度。

第二行为 N 个非负整数，按照文章的顺序，每个数（大小不超过 1000）代表一个英文单词。文章中两个单词是同一个单词，当且仅当它们对应的非负整数相同。

【输出】

输出文件 translate.out 共1行，包含一个整数，为软件需要查词典的次数。

【输入输出样例 1】

3 7

1 2 1 5 4 4 1

输出

5

【输入输出样例 1 说明】

整个查字典过程如下：每行表示一个单词的翻译，冒号前为本次翻译后的内存状况：

空：内存初始状态为空。

1． 1：查找单词1 并调入内存。

2． 1 2：查找单词 2 并调入内存。

3． 1 2：在内存中找到单词 1。

4． 1 2 5：查找单词 5 并调入内存。

5． 2 5 4：查找单词 4 并调入内存替代单词 1。

6． 2 5 4：在内存中找到单词 4。

7． 5 4 1：查找单词 1 并调入内存替代单词 2。

共计查了 5次词典。

【输入输出样例 2】

translate.in translate.out

2 10

8 824 11 78 11 78 11 78 8 264

6

【数据范围】

对于 10%的数据有 M=1，N≤ 5。

对于 100%的数据有 0<M≤ 100，0<N≤ 1000。

**NOIP2011 铺地毯（carpet.cpp/c/pas）**

【问题描述】为了准备一个独特的颁奖典礼，组织者在会场的一片矩形区域（可看做是平面直角坐标系的第一象限）铺上一些矩形地毯。一共有n 张地毯，编号从1 到n。

现在将这些地毯按照编号从小到大的顺序平行于坐标轴先后铺设，后铺的地毯覆盖在前面已经铺好的地毯之上。地毯铺设完成后，组织者想知道覆盖地面某个点的最上面的那张地毯的编号。

注意：在矩形地毯边界和四个顶点上的点也算被地毯覆盖。

【输入】

输入文件名为 carpet.in。

输入共 n+2 行。

第一行，一个整数 n，表示总共有n 张地毯。

接下来的 n 行中，第i+1 行表示编号i 的地毯的信息，包含四个正整数a，b，g，k，每两个整数之间用一个空格隔开，分别表示铺设地毯的左下角的坐标（a，b）以及地毯在x轴和y 轴方向的长度。

第 n+2 行包含两个正整数x 和y，表示所求的地面的点的坐标（x，y）。

【输出】

输出文件名为 carpet.out。

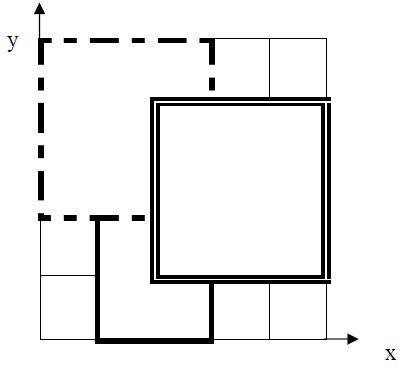
输出共 1 行，一个整数，表示所求的地毯的编号；若此处没有被地毯覆盖则输出-1。

【输入输出样例 1】

|  |  |
| --- | --- |
| carpet.in | carpet.out |
| 3  1 0 2 3  0 2 3 3  2 1 3 3  2 2 | 3 |

【输入输出样例说明】

如下图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，覆盖点（2，2）的最上面一张地毯是3 号地毯。



【输入输出样例 2】

|  |  |
| --- | --- |
| carpet.in | carpet.out |
| 3  1 0 2 3  0 2 3 3  2 1 3 3  4 5 | -1 |

【输入输出样例说明】

如上图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，点（4，5）没有被地毯覆盖，所以输出-1。

【数据范围】

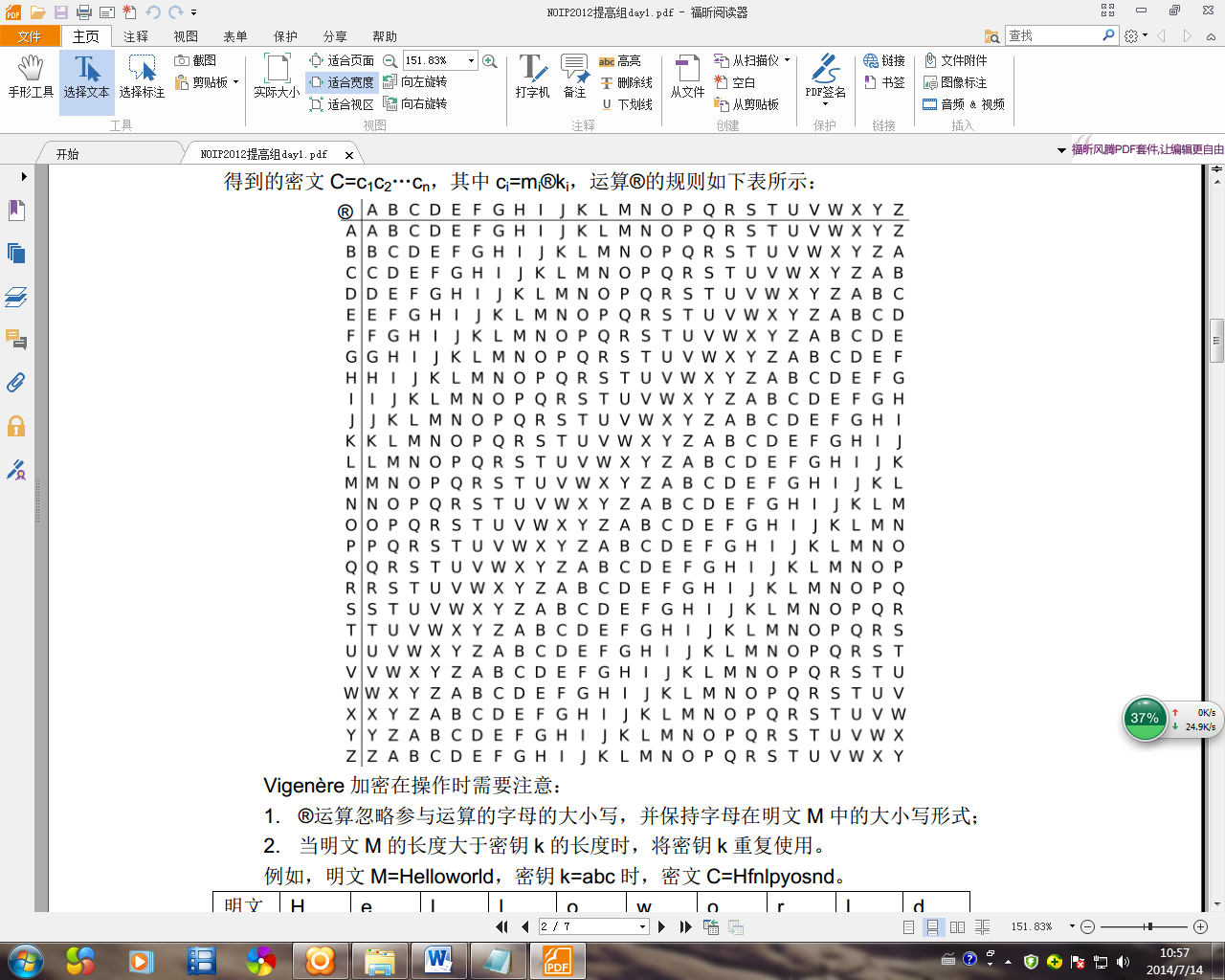
对于 30%的数据，有n≤2；

对于 50%的数据，0≤a, b, g, k≤100；

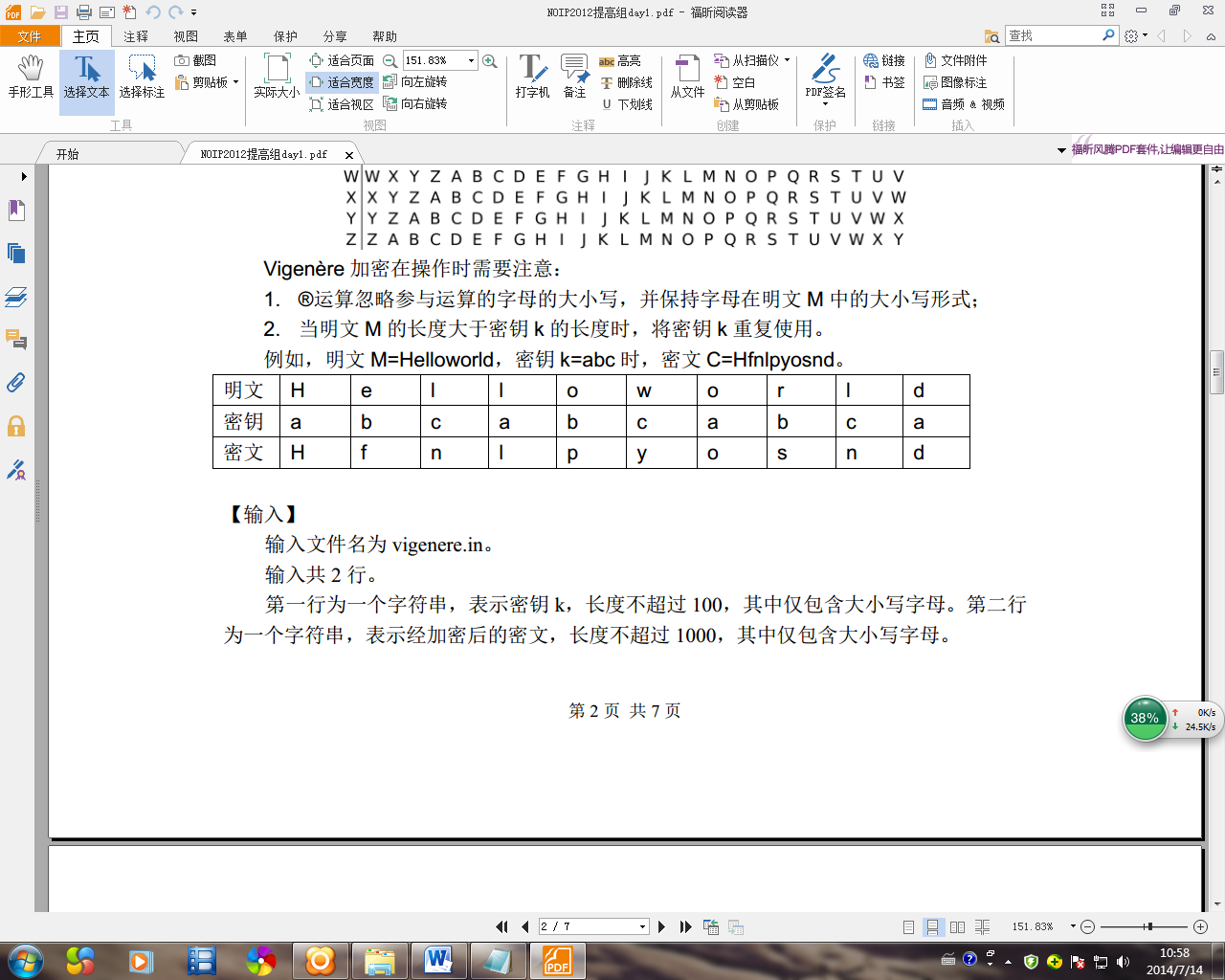
对于 100%的数据，有0≤n≤10,000，0≤a, b, g, k≤100,000。

**NOIP2012 Vigenère 密码 (vigenere.pas/c/cpp)**

【问题描述】 16 世纪法国外交家Blaise de Vigenère 设计了一种多表密码加密算法——Vigenère 密码。Vigenère 密码的加密解密算法简单易用，且破译难度比较高，曾在美国南北战争中为南军所广泛使用。 在密码学中，我们称需要加密的信息为明文，用 M 表示；称加密后的信息为密文，用C 表示；而密钥是一种参数， 是将明文转换为密文或将密文转换为明文的算法中输入的数据，记为 k。 在 Vigenère 密码中， 密钥 k 是一个字母串， k=k1k2…kn。当明文 M=m1m2…mn 时，得到的密文 C=c1c2…cn，其中 ci=mi®ki，运算®的规则如下表所示：



Vigenère 加密在操作时需要注意： 1. ®运算忽略参与运算的字母的大小写，并保持字母在明文M 中的大小写形式； 2. 当明文M 的长度大于密钥k 的长度时，将密钥k 重复使用。 例如，明文M=Helloworld，密钥k=abc 时，密文C=Hfnlpyosnd。



【输入】

输入文件名为 vigenere.in。

输入共 2 行。

第一行为一个字符串，表示密钥 k，长度不超过 100，其中仅包含大小写字母。

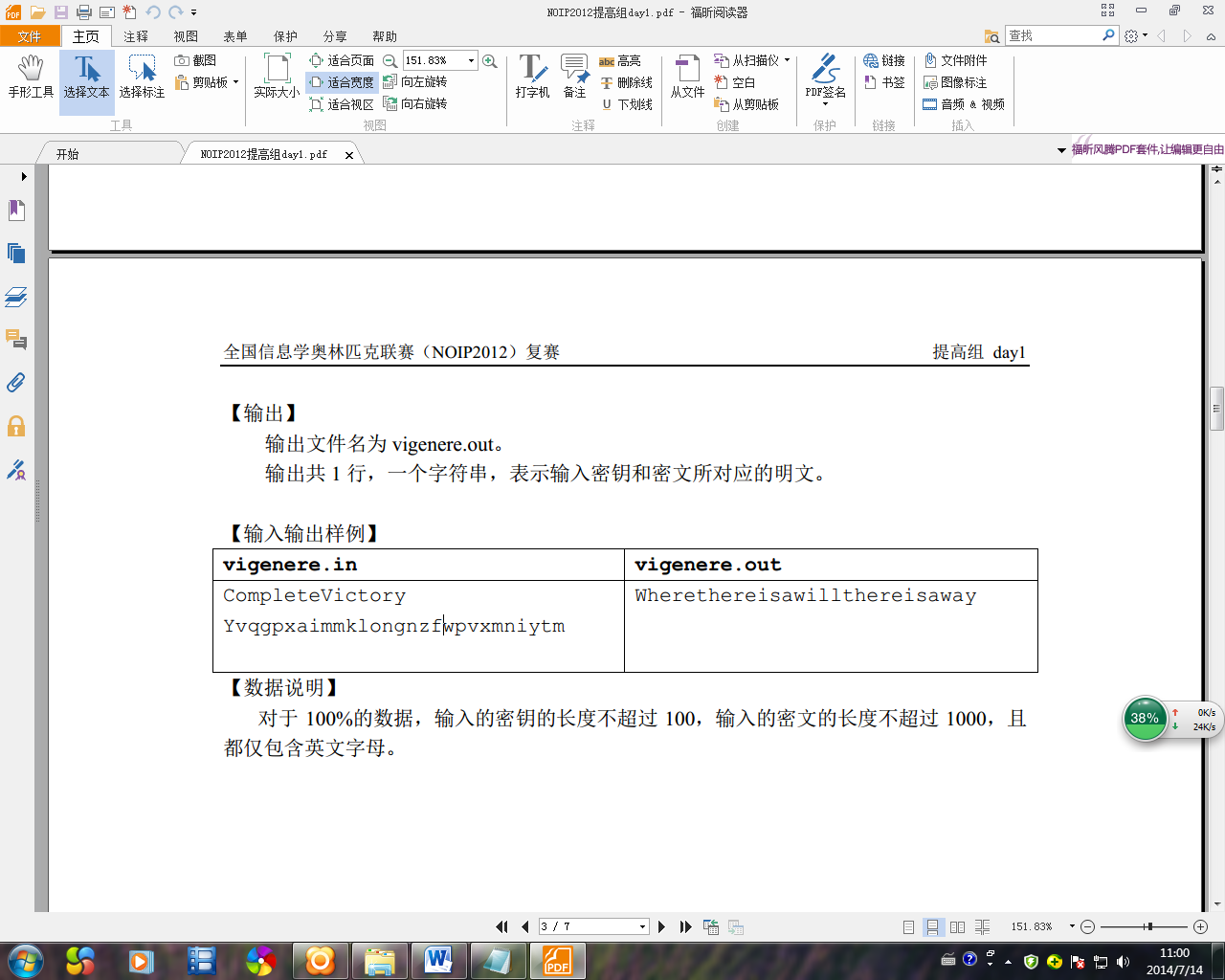
第二行为一个字符串，表示经加密后的密文，长度不超过 1000，其中仅包含大小写字母。

【输出】

输出文件名为 vigenere.out。

输出共 1 行，一个字符串，表示输入密钥和密文所对应的明文。

【输入输出样例】



【数据说明】

对于 100%的数据，输入的密钥的长度不超过 100，输入的密文的长度不超过 1000，且都仅包含英文字母。

**NOIP2013 转圈游戏(circle.cpp/c/pas)**

【问题描述】n 个小伙伴（编号从 0 到 n-1）围坐一圈玩游戏。按照顺时针方向给 n 个位置编号，从0 到 n-1。最初，第 0 号小伙伴在第 0 号位置，第 1 号小伙伴在第 1 号位置，……，依此类推。

游戏规则如下：每一轮第 0 号位置上的小伙伴顺时针走到第 m 号位置，第 1 号位置小伙伴走到第 m+1 号位置，……，依此类推，第n − m号位置上的小伙伴走到第 0 号位置，第n-m+1 号位置上的小伙伴走到第 1 号位置，……，第 n-1 号位置上的小伙伴顺时针走到第m-1 号位置。

现在，一共进行了 10k轮，请问 x 号小伙伴最后走到了第几号位置。

【输入】

输入文件名为 circle.in。

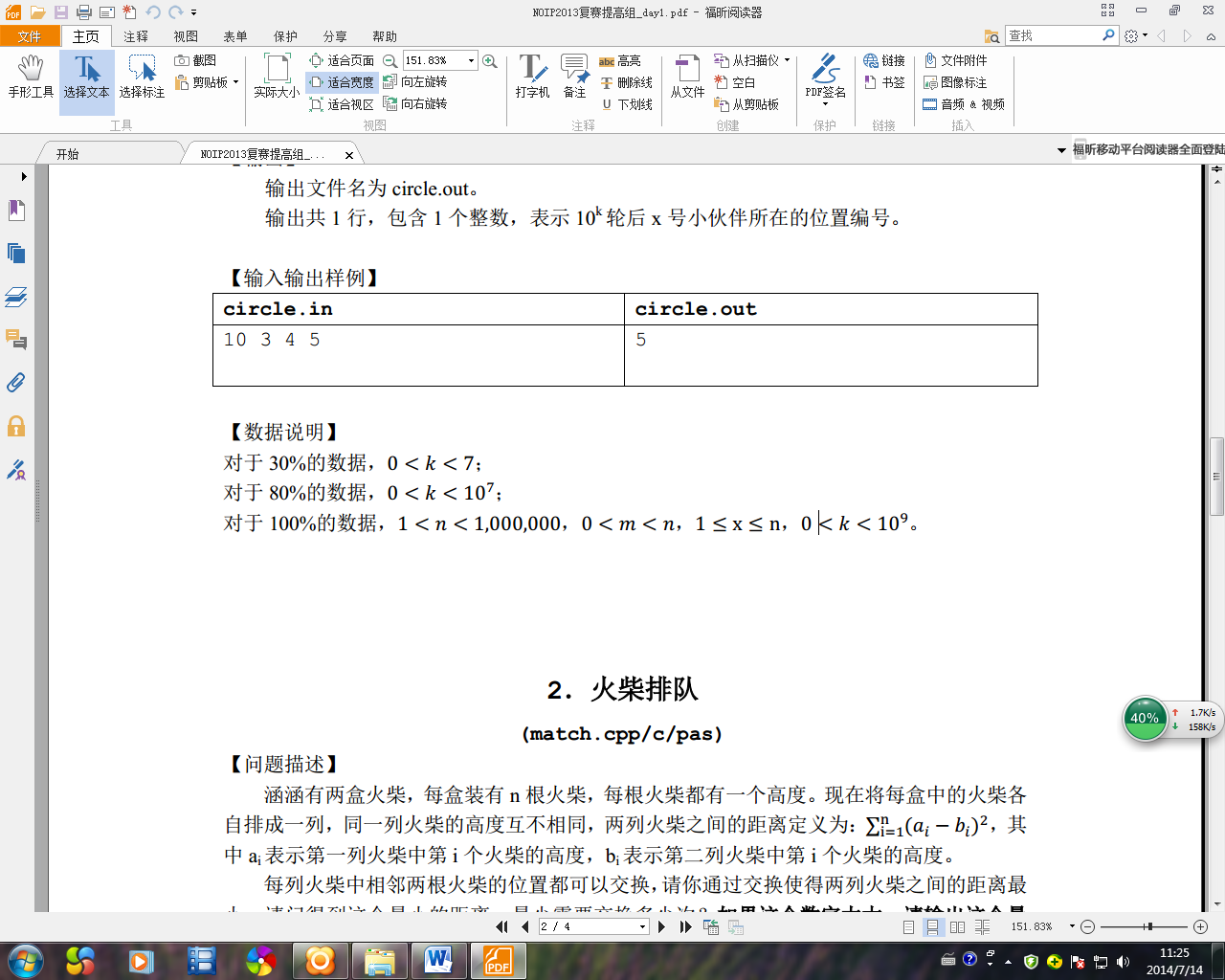
输入共 1 行，包含 4 个整数 n、m、k、x，每两个整数之间用一个空格隔开。

【输出】

输出文件名为 circle.out。

输出共 1 行，包含 1 个整数，表示 10k轮后 x 号小伙伴所在的位置编号。

【输入输出样例】



【数据说明】

对于 30%的数据，0 < 𝑘 < 7；

对于 80%的数据，0 < 𝑘 < 10^7；

对于 100%的数据，1 < 𝑛 < 1,000,000，0 < 𝑚 < 𝑛，1 ≤ x ≤ n，0 < 𝑘 < 10/69。

**NOIP2014 生活大爆炸版石头剪刀布 (rps.cpp/c/pas)**

【问题描述】

石头剪刀布是常见的猜拳游戏：石头胜剪刀，剪刀胜布，布胜石头。如果两个人出拳一 样，则不分胜负。在《生活大爆炸》第二季第 8 集中出现了一种石头剪刀布的升级版游戏。 升级版游戏在传统的石头剪刀布游戏的基础上，增加了两个新手势：

斯波克：《星际迷航》主角之一。 蜥蜴人：《星际迷航》中的反面角色。

这五种手势的胜负关系如表一所示，表中列出的是甲对乙的游戏结果。

表一 石头剪刀布升级版胜负关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 乙 甲对乙的  甲 结果 | 剪刀 | 石头 | 布 | 蜥蜴人 | 斯波克 |
| 剪刀 | 平 | 输 | 赢 | 赢 | 输 |
| 石头 |  | 平 | 输 | 赢 | 输 |
| 布 |  |  | 平 | 输 | 赢 |
| 蜥蜴人 |  |  |  | 平 | 赢 |
| 斯波克 |  |  |  |  | 平 |

现在，小 A 和小 B 尝试玩这种升级版的猜拳游戏。已知他们的出拳都是有周期性规律的， 但周期长度不一定相等。例如：如果小 A 以“石头-布-石头-剪刀-蜥蜴人-斯波克”长度为

6 的周期出拳，那么他的出拳序列就是“石头-布-石头-剪刀-蜥蜴人-斯波克-石头-布-石头

-剪刀-蜥蜴人-斯波克-……”，而如果小 B 以“剪刀-石头-布-斯波克-蜥蜴人”长度为 5 的 周期出拳，那么他出拳的序列就是“剪刀-石头-布-斯波克-蜥蜴人-剪刀-石头-布-斯波克- 蜥蜴人-……”

已知小 A 和小 B 一共进行 N 次猜拳。每一次赢的人得 1 分，输的得 0 分；平局两人都得 0 分。现请你统计 N 次猜拳结束之后两人的得分。

【输入】

输入文件名为 rps.in。

第一行包含三个整数：N，NA，NB，分 别 表 示 共 进 行 N 次猜拳、小 A 出拳的周期 长度，小 B 出拳的周期长度。数与数之间以一个空格分隔。

第二行包含 NA 个整数，表示小 A 出拳的规律，第三行包含 NB 个整数，表示小 B 出拳的 规律。其中，0 表示“剪刀”，1 表示“石头”，2 表示“布”，3 表示“蜥蜴人”， 4 表示“斯 波克”。数与数之间以一个空格分隔。

【输出】

输出文件名为 rps.out。

输出一行， 包含两个整数，以一个空格分隔，分别表示小 A、小 B 的得分。

【输入输出样例 1】

|  |  |
| --- | --- |
| rps.in | rps.out |
| 10 5 6  0 1 2 3 4  0 3 4 2 1 0 | 6 2 |

【输入输出样例 2】

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| rps.in | | | | | rps.out |
| 9 | 5 | 5 |  |  | 4 4 |
| 0 | 1 | 2 | 3 | 4 |
| 1 | 0 | 3 | 2 | 4 |

【数据说明】

对于 100%的数据，0 < N ≤ 200，0 < NA ≤ 200， 0 < NB ≤ 200。

**NOIP2015神奇的幻方(magic.cpp/c/pas)**

【问题描述】幻方是一种很神奇的 𝑁 ∗ 𝑁 矩阵：它由数字 1,2,3, … … , 𝑁 ∗ 𝑁 构成，且每行、每列

及两条对角线上的数字之和都相同。

当 𝑁 为奇数时，我们可以通过以下方法构建一个幻方： 首先将 1 写在第一行的中间。

之后，按如下方式从小到大依次填写每个数 𝐾(𝐾 = 2,3, … , 𝑁 ∗ 𝑁) ：

1. 若 (𝐾 − 1) 在第一行但不在最后一列，则将 𝐾 填在最后一行， (𝐾 − 1) 所在列

的右一列；

2. 若 (𝐾 − 1) 在最后一列但不在第一行，则将 𝐾 填在第一列，(𝐾 − 1) 所在行的上

一行；

3. 若 (𝐾 − 1) 在第一行最后一列，则将 𝐾 填在 (𝐾 − 1) 的正下方；

4. 若 (𝐾 − 1) 既不在第一行，也不在最后一列，如果 (𝐾 − 1) 的右上方还未填数， 则将 K 填在(𝐾 − 1)的右上方，否则将 𝐾 填在 (𝐾 − 1) 的正下方。

现给定 𝑁，请按上述方法构造 𝑁 ∗ 𝑁 的幻方。

【输入】

输入文件名为 magic.in。

输入文件只有一行，包含一个整数 𝑁，即幻方的大小。

【输出】

输出文件名为 magic.out。

输出文件包含 𝑁 行，每行 𝑁 个整数，即按上述方法构造出的 𝑁 ∗ 𝑁 的幻方。相邻 两个整数之间用单个空格隔开。

【输入输出样例 1】

|  |  |
| --- | --- |
| magic.in | magic.out |
| 3 | 8 1 6  3 5 7  4 9 2 |

见选手目录下的 magic/magic1.in 和 magic/magic1.ans。

【输入输出样例 2】

见选手目录下的 magic/magic2.in 和 magic/magic2.ans。

【数据说明】

对于 100% 的数据，1 ≤ 𝑁 ≤ 39 且 𝑁 为奇数。

**NOIP2016玩具谜题（toy.cpp/c/pas)**

【问题描述】小南有一套可爱的玩具小人，它们各有不同的职业。 有一天，这些玩具小人把小南的眼镜藏了起来。 小南发现玩具小人们围成了一个圈，它们有的面朝圈内，有的面朝圈外。

如下图：



这时 singer 告诉小南一个谜题：“眼镜藏在我左数第 3 个玩具小人的右数第 1 个玩 具小人的左数第 2 个玩具小人那里。 ”

小南发现，这个谜题中玩具小人的朝向非常关键，因为朝内和朝外的玩具小人的 左右方向是相反的：面朝圈内的玩具小人，它的左边是顺时针方向，右边是逆时针方 向；而面向圈外的玩具小人，它的左边是逆时针方向，右边是顺时针方向。

小南一边艰难地辨认着玩具小人，一边数着： “singer 朝内，左数第 3 个是 archer。 “archer 朝外，右数第 1 个是 thinker。 “thinker 朝外，左数第 2 个是 writer。 “所以 眼镜藏在 writer 这里！ ”

虽然成功找回了眼镜，但小南并没有放心。 如果下次有更多的玩具小人藏他的 眼 镜，或是谜题的长度更长，他可能就无法找到眼镜了。 所以小南希望你写程序 帮他解 决类似的谜题。 这样的谜题具体可以描述为：

有 n 个玩具小人围成一圈，己知它们的职业和朝向。 现在第 1 个玩具小人告诉小 南一个包含 m 条指令的谜题，其中第 i 条指令形如“左数/右数第 si 个玩具小人”。 你需要输出依次数完这些指令后，到达的玩具小人的职业。

【输入格式】

从文件 toy.in 中读入数据。

输入的第一行包含两个正整数 n,m ，表示玩具小人的个数和指令的条数。 接下来 n 行，每行包含一个整数和一个字符串，以逆时针为顺序给出每个玩具小

人的朝向和职业。其中 0 表示朝向圈内， 1 表示朝向圈外。保证不会出现其他的数。字 符串长度不超过 10 且仅由小写字母构成，字符串不为空，并且字符串两两不同。 整数 和字符串之间用一个空格隔开。

接下来 m 行，其中第 i 行包含两个整数 ai, si ，表示第 i 条指令。若 ai = 0 ，表示向 左数 si 个人；若 ai = 1 ，表示向右数 si 个人。保证 ai 不会出现其他的数， 1 ≤ si < n 。

【输出格式】

输出到文件 toy.out 中。 输出一个字符串，表示从第一个读入的小人开始，依次数 完 m条指令后到达的小人的职业。

【样例 1 输入】

7 3

0 singer

0 reader

0 mengbier

1 thinker

1 archer

0 writer

1 mogician

0 3

1 1

0 2

【样例 1 输出】

writer

【样例 1 说明】

这组数据就是【题目描述】中提到的例子。

【样例 2 输入】

10 10

1 c

0 r

0 p

1 d

1 e

1 m

1 t

1 y

1 u

0 v

1 7

1 1

1 4

0 5

0 3

0 1

1 6

1 2

0 8

0 4

【样例 2 输出】

y

【子任务】

子任务会给出部分测试数据的特点。如果你在解决题目中遇到了困难，可以尝试 只解决一部分测试数据。

每个测试点的数据规模及特点如下表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试点 | *n* | *m* | 全朝内 | 全左数 | *si* = 1 | 职业长度为 1 |
| 1 | = 20 | = 103 | √ | √ | √ | √ |
| 2 | × |
| 3 | √ | × |
| 4 | × |
| 5 | √ | √ | × |
| 6 | × |
| 7 | √ | × |
| 8 | × |
| 9 | √ | √ | √ | × |
| 10 | × |
| 11 | √ | × |
| 12 | × |
| 13 | √ | √ | × |
| 14 | × |
| 15 | √ | × |
| 16 | × |
| 17 | = 105 | = 105 | √ | √ |
| 18 | × |
| 19 | √ | × |
| 20 | × |

其中一些简写的列意义如下：

• 全朝内：若为“√”，表示该测试点保证所有的玩具小人都朝向圈内；

• 全左数： 若为“√”， 表示该测试点保证所有的指令都向左数， 即对任意 的 1 ≤ i ≤ m ， ai = 0 ；

• si = 1 ：若为“√”，表示该测试点保证所有的指令都只数 1 个，即对任意的

1 ≤ i ≤ m ， si = 1 ；

• 职业长度为 1 ：若为“√”，表示该测试点保证所有玩具小人的职业一定是一 个长度为 1 的字符串。

**NOIP2017 小凯的疑惑(math.cpp/c/pas)**

【问题描述】小凯手中有两种面值的金币，两种面值均为正整数且彼此互素。每种金币小凯都有 无数个。在不找零的情况下，仅凭这两种金币，有些物品他是无法准确支付的。现在小 凯想知道在无法准确支付的物品中，最贵的价值是多少金币？注意：输入数据保证存在 小凯无法准确支付的商品。

【输入格式】

输入文件名为 math.in。

输入数据仅一行，包含两个正整数 a 和 b，它们之间用一个空格隔开，表示小凯手 中金币的面值。

【输出格式】

输出文件名为 math.out。

输出文件仅一行，一个正整数 N，表示不找零的情况下，小凯用手中的金币不能准 确支付的最贵的物品的价值。

【输入输出样例 1】

|  |  |
| --- | --- |
| math.in | math.out |
| 3 7 | 11 |

见选手目录下的 math/math1.in 和 math/math1.ans。

【输入输出样例 1 说明】

小凯手中有面值为 3 和 7 的金币无数个，在不找零的前提下无法准确支付价值为 1、 2、4、5、8、11 的物品，其中最贵的物品价值为 11，比 11 贵的物品都能买到，比如：

12 = 3 \* 4 + 7 \* 0

13 = 3 \* 2 + 7 \* 1

14 = 3 \* 0 + 7 \* 2

15 = 3 \* 5 + 7 \* 0

……

【输入输出样例 2】

见选手目录下的 math/math2.in 和 math/math2.ans。

【数据规模与约定】

对于 30%的数据： 1 ≤ a，b ≤ 50。

对于 60%的数据： 1 ≤ a，b ≤ 10,000。

对于 100%的数据：1 ≤ a，b ≤ 1,000,000,000。

**NOIP2018 铺设道路(road.cpp/c/pas)**

【问题描述】

春春是一名道路工程师，负责铺设一条长度为 n 的道路。 铺设道路的主要工作是填平下陷的地表。整段道路可以看作是 n 块首尾相连的区

域，一开始，第 i 块区域下陷的深度为 di 。

春春每天可以选择一段连续区间 [L, R] ，填充这段区间中的每块区域，让其下陷深 度减少 1。在选择区间时，需要保证，区间内的每块区域在填充前下陷深度均不为 0 。

春春希望你能帮他设计一种方案，可以在最短的时间内将整段道路的下陷深度都变

为 0 。

【输入格式】

输入文件名为 road.in。

输入文件包含两行，第一行包含一个整数 n，表示道路的长度。 第二行包含 n 个整数，相邻两数间用一个空格隔开，第 i 个整数为 di 。

【输出格式】

输出文件名为 road.out。 输出文件仅包含一个整数，即最少需要多少天才能完成任务。

【输入输出样例 1】

|  |  |
| --- | --- |
| road.in | road.out |
| 6  4 3 2 5 3 5 | 9 |

见选手目录下的 road/road1.in 和 road/road1.ans。

【样例解释】

一种可行的最佳方案是，依次选择：

[1,6]、[1,6]、[1,2]、[1,1]、[4,6]、[4,4]、[4,4]、[6,6]、[6,6]。

【输入输出样例 2】

见选手目录下的 road/road2.in 和 road/road2.ans。

【数据规模与约定】

对于 30% 的数据，1 ≤ � ≤ 10 ； 对于 70% 的数据，1 ≤ � ≤ 1000 ；

对于 100% 的数据，1 ≤ � ≤ 100000 ，0 ≤ di ≤ 10000 。