

上机训练题

引例： 字符串的读入和输出（`00read`, `read.in`, `read.out`）

问题描述：读入下列字符串，然后每个字符串一行输出。

输入：第一行是用空格隔开的多个字符串；

第二行是用逗号隔开的多个字符串；

第三行开头是一个整数 n ，接下来有 n 个用空格隔开的字符串；

第四行开头是一个整数 n ，接下来有 n 个用逗号隔开的字符串；

第五行是一个整数 n ；

从第六行开始每行就是一个字符串，共有 n 行。

输出：每个字符串一行，不用输出字符串的个数 n 以及逗号等字符串之间的间隔符号，最后一行之后要有一个回车符。

样例输入：

abc efdg

dgh,erty,dshhs

2 dgh ertya

3 dgh,erty,dshhss

2

"asdfasgyrterue dwfgdfhtyw"

"pouuurty rwe{[]}#@!asdfasduuioio"

样例输出：

abc

efdg

dgh

erty

dshhs

dgh

ertya

dgh

erty

dshhss

"asdfasgyrterue dwfgdfhtyw"

"pouuurty rwe{[]}#@!asdfasduuioio"

一、上机训练题目

例 1 字符和整数之间的通用性 (01trans, trans.in, trans.out)

问题描述：字符用ascii码赋值，将字符进行强制类型转化，转化成对应的整数，输出。(01trans, trans.in, trans.out)

样例输入：65

样例输出：A 65

例 2 任意大小写字母转化 (02trans, trans.in, trans.out)

问题描述：输入一串字符以'#'结束，大写字母转化成小写，小写字母转化成大写。

样例输入：abAB#

样例输出：ABab

例3、字母表输出（03char, char.in, char.out）

问题描述：按字母表顺序和逆序每隔一个字母打印。即打印出：

样例输出：

a c e g i k m o q s u w y

z x r v t p n l j h f d b

例4、输出字符串（04string, string.in, string.out）

问题描述：输入一个字符串，分别输出字符串中的每一个字符

样例输入：abc

样例输出：a b c

例5、数字统计（05count, count.in, count.out）

问题描述：（请使用sprintf()函数）

请统计某个给定范围[L, R]的所有整数中，数字2 出现的次数。

比如给定范围[2, 22]，数字2 在数2 中出现了1 次，在数12 中出现1 次，在数20 中出现1 次，在数21 中出现1 次，在数22 中出现2 次，所以数字2 在该范围内一共出现了6 次。

样例输入：2 22

样例输出：6

例6、数字翻转（06turn, turn.in, turn.out）

问题描述：（请使用scanf()和sprintf()函数）

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应

满足整数的常见形式，即除给定的原数为零，否则反转后

得到的新数的最高位数字不应为零

样例输入：-380

样例输出：-83

例7、单词统计（07count, count.in, count.out）

问题描述：一般的文本编辑器都有查找单词的功能，该功能可以快速定位特定单词在文章中的位置，有的还能统计出特定单词在文章中出现的次数。

现在，请你编程实现这一功能，具体要求是：给定一个单词，请你输出它在给定的文章中出现的次数和第一次出现的位置。注意：匹配单词时，不区分大小写，但要求完全匹配，即给定单词必须与文章中的某一独立单词在不区分大小写的情况下完全相同（参见样例1），如果给定单词仅是文章中某一单词的一部分则不算匹配（参见样例2）。

【输入】

第1 行为一个字符串，其中只含字母，表示给定单词；

第2 行为一个字符串，其中只可能包含字母和空格，表示给定的文章。

【输出】

只有一行，如果在文章中找到给定单词则输出两个整数，两个整数之间用一个空格隔开，分别是单词在文章中出现的次数和第一次出现的位置（即在文章中第一次出现时，单词首字母在文章中的位置，位置从0 开始）；如单词在文章中没有出现，则直接输出一个整数-1。

【输入样例1】

To

to be or not to be is a question

【输出样例1】

2 0

【输入输出样例1说明】 输出结果表示给定的单词To 在文章中出现两次，第一次出现的位置为0。

【数据范围】

$1 \leq \text{单词长度} \leq 10$ 。

$1 \leq \text{文章长度} \leq 1,000,000$ 。

例8、回文字符串（08plalind, plalind.in, plalind.out）

问题描述：输入一个字符串，输出该字符串是否回文。回文是指顺读和倒读都一样的字符串。

输入:输入为一行字符串（字符串中没有空白字符，字符串长度不超过100）。

输出:如果字符串是回文，输出yes；否则，输出no。

样例输入:abcdedcba

样例输出:yes

二、拓展提升题目：

1.统计数字字符个数(09count, count.in, count.out)

总时间限制:

1000ms

内存限制:

65536kB

描述

输入一行字符，统计出其中数字字符的个数。

输入

一行字符串，总长度不超过255。

输出

输出为1行，输出字符串里面数字字符的个数。

样例输入

Peking University is set up at 1898.

样例输出

4

2.简单密码（10password, password.in, password.out）

总时间限制:

1000ms

内存限制:

65536kB

描述

Julius Caesar曾经使用过一种很简单的密码。对于明文中的每个字

符，将它用它字母表中后5位对应的字符来代替，这样就得到了密文。

比如字符A用F来代替。如下是密文和明文中字符的对应关系。

密文

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

明文

V W X Y Z A B C D E F G H I J K L M N O P Q R S T U

你的任务是对给定的密文进行解密得到明文。

你需要注意的是，密文中出现的字母都是大写字母。密文中也包括非字母的字符，对这些字符不用进行解码。

输入

一行，给出密文，密文不为空，而且其中的字符数不超过200。

输出

输出一行，即密文对应的明文。

样例输入

NS BFW, JAJSYX TK NRUTWYFSHJ FWJ YMJ WJXZQY TK
YWNANFQ HFZXJX

样例输出

IN WAR, EVENTS OF IMPORTANCE ARE THE RESULT OF
TRIVIAL CAUSES

3.基因相关性(11jene, jene.in, jene.out)

总时间限制:

1000ms

内存限制:

65536kB

描述

为了获知基因序列在功能和结构上的相似性，经常需要将几条不同序列的 **DNA** 进行比对，以判断该比对的 **DNA** 是否具有相关性。

现比对两条长度相同的 **DNA** 序列。首先定义两条 **DNA** 序列相同位置的碱基为一个碱基对，如果一个碱基对中的两个碱基相同的话，则称为相同碱基对。接着计算相同碱基对占总碱基对数量的比例，如果该比例大于等于给定阈值时则判定该两条 **DNA** 序列是相关的，否则不相关。

输入

有三行，第一行是用来判定出两条 **DNA** 序列是否相关的阈值，随后 2 行是两条 **DNA** 序列（长度不大于 500）。

输出

若两条 **DNA** 序列相关，则输出“yes”，否则输出“no”。

样例输入

0.85

ATCGCCGTAAGTAACGGTTTTAAATAGGCC

ATCGCCGGAAGTAACGGTCTTAAATAGGCC

样例输出

yes

4.删除单词后缀（12delete, delete.in, delete.out）

总时间限制:

1000ms

内存限制:

65536kB

描述

给定一个单词，如果该单词以 **er**、**ly** 或者 **ing** 后缀结尾， 则删除该后缀（题目保证删除后缀后的单词长度不为 0）， 否则不进行任何操作。

输入

输入一行，包含一个单词（单词中间没有空格，每个单词最大长度为 32）。

输出

输出按照题目要求处理后的单词。

样例输入

referer

样例输出

refer

5.石头剪子布（13stone, stone.in, stone.out）

总时间限制:

1000ms

内存限制:

65536kB

描述

石头剪子布，是一种猜拳游戏。起源于中国，然后传到日本、朝鲜等地，随着亚欧贸易的不断发展它传到了欧洲，到了近现代逐渐风靡世界。简单明了的规则，使得石头剪子布没有任何规则漏洞可钻，单次玩法比拼运气，多回合玩法比拼心理博弈，使得石头剪子布这个古老的游戏同时用于“意外”与“技术”两种特性，深受世界人民喜爱。

游戏规则：石头打剪刀，布包石头，剪刀剪布。

现在，需要你写一个程序来判断石头剪子布游戏的结果。

输入

输入包括 $N+1$ 行：

第一行是一个整数 N ，表示一共进行了 N 次游戏。 $1 \leq N \leq 100$ 。

接下来 N 行的每一行包括两个字符串，表示游戏参与者 Player1，

Player2 的选择（石头、剪子或者是布）：

S1 S2

字符串之间以空格隔开 S1,S2 只可能取值在{"Rock", "Scissors", "Paper"}（大小写敏感）中。

输出

包括 N 行，每一行对应一个胜利者（Player1 或者 Player2），或者游戏出现平局，则输出 Tie。

样例输入

3

Rock Scissors

Paper Paper

Rock Paper

样例输出

Player1

Tie

Player2

提示

Rock 是石头，Scissors 是剪刀，Paper 是布。

6.单词翻转(14turn, turn.in, turn.out)

总时间限制:

1000ms

内存限制:

65536kB

描述

输入一个句子（一行），将句子中的每一个单词翻转后输出。

输入

只有一行，为一个字符串，包含大小写字母和空格，不超过 500 个字符。单词之间以空格隔开。

输出

翻转每一个单词后的字符串，单词之间的空格需与原文一致。

样例输入

hello world

样例输出

olleh dlrow

7、演示将美国格式的日期转换为国际格式的例子。（**15date, date.in, date.out**）

美国使用月日年格式例如 May 28 , 2002。

转为国际格式应为 28 May 2002。

这可以使用 **find** 检索，使用 **substr** 截取，然后再按要求重组

具体步骤如下：

① 寻找月份。

假设 `string` 类的 `Date` 对象存放美国日期，则下列语句

```
int i=Date.find( " " ); //查找 May 与 28 之间的空格，i=3
```

```
string Month=Date.substr(0,i); //从 0 位置开始截取 i 个字符
```

(上面求的 $i=3$)得到 `Month`。

② 寻找日子。

寻找 “,” 号，假设其位置为 k 。`Day` 的起始位置应是 $i+1$ 处，长度应是 $k-(i+1)$ ，则语句

```
string Day=Date.substr(i+1, k-i-1);          得到 Day。
```

③ 寻找年份。

年份应从逗号处再计数 2 次（因逗号后面有一个空格），长度为 4，也可使用比年的长度长的字符串的总长度，这由它的成员函数 `size` 提供。下面两种方法是等效的：

```
string Year = Date.substr(k + 2, Date.size() - 1);
```

```
string Year = Date.substr(k + 2, 4);
```

④ 按新格式拼装并输出。

样例输入：

May 28, 2002

样例输出：

28 May 2002

8、单词替换 (16replace,replace.in,replace.out)

时间限制: 1000 ms 内存限制: 65536 KB

【题目描述】输入一个字符串，以回车结束（字符串长度 ≤ 200 ）。

该字符串由若干个单词组成，单词之间用一个空格隔开，所有单词区分大小写。现需要将其中的某个单词替换成另一个单词，并输出替换之后的字符串。

【输入】第 1 行是包含多个单词的字符串 s ;

第 2 行是待替换的单词 a (长度 ≤ 100);

第 3 行是 a 将被替换的单词 b (长度 ≤ 100)。 s, a, b 最前面和最后面都没有空格。

【输出】输出只有 1 行，将 s 中所有单词 a 替换成 b 之后的字符串。

【输入样例】

You want someone to help you

You

I

【输出样例】

I want someone to help you

9、小书童——密码 (17code, code.in, code.out)

题目描述

小牛迷上了“小书童”，有一天登陆时忘记密码了（他没绑定邮箱 or 手机），于是便把问题抛给了神犇你。小牛虽然忘记密码，但他还记得密码是由一个字符串组成。密码是由原文字符串（由不超过 50 个小写字母组成）中每个字母向后移动 n 位形成的。 z 的下一个字

母是 **a**，如此循环。他现在找到了移动前的原文字符串及 **n**，请你求出密码。

输入格式

第一行：**n**。第二行：未移动前的一串字母

输出格式

一行，是此前的密码

样例输入：

1

qwe

样例输出：

rxf

字符串长度 ≤ 50

10、你的飞碟在这儿 **Your Ride Is He...** (18ufo, ufo.in, ufo.out)

题目描述

众所周知，在每一个彗星后都有一只 **UFO**。这些 **UFO** 时常来收集地球上的忠诚支持者。不幸的是，他们的飞碟每次出行都只能带上一组支持者。因此，他们要用一种聪明的方案让这些小组提前知道谁会被彗星带走。他们为每个彗星起了一个名字，通过这些名字来决定这个小组是不是被带走的那个特定的小组（你认为谁给这些彗星取的名字呢？）。关于如何搭配的细节会在下面告诉你；你的任务是写一个

程序,通过小组名和彗星名来决定这个小组是否能被那颗彗星后面的 UFO 带走。

小组名和彗星名都以下列方式转换成一个数字:最终的数字就是名字中所有字母的积,其中 A 是 1, Z 是 26。例如,USACO 小组就是 $21 \times 19 \times 1 \times 3 \times 15 = 17955$ 。如果小组的数字 $\text{mod } 47$ 等于彗星的数字 $\text{mod } 47$,你就得告诉这个小组需要准备好被带走!(记住“ $a \text{ mod } b$ ”是 a 除以 b 的余数; $34 \text{ mod } 10$ 等于 4)

写出一个程序,读入彗星名和小组名并算出用上面的方案能否将两个名字搭配起来,如果能搭配,就输出“GO”,否则输出“STAY”。小组名和彗星名均是没有空格或标点的一串大写字母(不超过 6 个字母)。

输入格式

第 1 行:一个长度为 1 到 6 的大写字母串,表示彗星的名字。

第 2 行:一个长度为 1 到 6 的大写字母串,表示队伍的名字。

输出格式

无

样例输入 1 :

COMETQ

HVNGAT

样例输出 1:

GO

样例输入 2:

ABSTAR

USACO

样例输出 2:

STAY

11、字符串最大跨距 (19span, span.in, span.out)

有三个字符串 S, S1, S2, 其中, S 长度不超过 300, S1 和 S2 的长度不超过 10。想检测 S1 和 S2 是否同时在 S 中出现, 且 S1 位于 S2 的左边, 并在 S 中互不交叉 (即, S1 的右边界点在 S2 的左边界点的左侧)。计算满足上述条件的最大跨距 (即, 最大间隔距离: 最右边的 S2 的起始点与最左边的 S1 的终止点之间的字符数目)。如果没有满足条件的 S1, S2 存在, 则输出 -1。

例如, S = "abcd123ab888efghij45ef67kl", S1="ab", S2="ef", 其中, S1 在 S 中出现了 2 次, S2 也在 S 中出现了 2 次, 最大跨距为: 18。
输入三个串: S1, S2, S3, 其间以逗号间隔 (注意, S1, S2, S3 中均不含逗号和空格);

输出 S1 和 S2 在 S 最大跨距; 若在 S 中没有满足条件的 S1 和 S2, 则输出 -1。

样例输入:

abcd123ab888efghij45ef67kl,ab,ef

样例输出:

12、ISBN 码 (20isbn, isbn.in, isbn.out)

每一本正式出版的图书都有一个 ISBN 号码与之对应，ISBN 码包括 9 位数字、1 位识别码和 3 位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”是分隔符（键盘上的减号），最后一位是识别码，例如 0-670-82162-4 就是一个标准的 ISBN 码。ISBN 码的首位数字表示书籍的出版语言，例如 0 代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如 670 代表维京出版社；第二个分隔之后的五位数字代表该书在出版社的编号；最后一位为识别码。识别码的计算方法如下：首位数字乘以 1 加上次位数字乘以 2……以此类推，用所得的结果 mod 11，所得的余数即为识别码，如果余数为 10，则识别码为大写字母 X。例如 ISBN 号码 0-670-82162-4 中的识别码 4 是这样得到的：对 067082162 这 9 个数字，从左至右，分别乘以 1,2,⋯,9，再求和，即 $0 \times 1 + 6 \times 2 + \dots + 2 \times 9 = 158$ ，然后取 $158 \bmod 11$ 的结果 4 作为识别码。你的任务是编写程序判断输入的 ISBN 号码中识别码是否正确，如果正确，则仅输出“Right”；如果错误，则输出你认为是正确的 ISBN 号码。

输入只有一行，是一个字符序列，表示一本书的 ISBN 号码（保证输入符合 ISBN 号码的格式要求）。输出共一行，假如输入的 ISBN 号码的识别码正确，那么输出“Right”，否则，按照规定的格式，输出正确的 ISBN 号码（包括分隔符“-”）。

样例输入

样例 #1: 0-670-82162-4

样例 #2: 0-670-82162-0

样例输出

样例 #1: Right

样例 #2: 0-670-82162-4

13、字符环（21ring, ring.in, ring.out）

总时间限制: 1000ms 内存限制: 65536kB

描述

有两个由字符构成的环。请写一个程序，计算这两个字符环上最长连续公共字符串的长度。例如，字符串“ABCEFAGADEGKABUVKLM”的首尾连在一起，构成一个环；字符串“MADJKLUVKL”的首尾连在一起，构成一个另一个环；“UVKLMA”是这两个环的一个连续公共字符串。

输入 一行，包含两个字符串，分别对应一个字符环。这两个字符串之间用单个空格分开。字符串长度不超过 255，且不包含空格等空白符。

输出 输出一个整数，表示这两个字符环上最长公共字符串的长度。

样例输入：

ABCEFAGADEGKABUVKLM MADJKLUVKL

样例输出：

