

A 告别的字符画

题目描述

这是最后一次上机啦！挥挥手和纯狐告别吧！

请输出告别的字符画吧。

Hint

善用搜索-替换功能

在 `C` 语言的字符串中需要转换的转义字符有：

输出 `"` 需要使用 `\`

输出 `'` 需要使用 `\`

输出 `\` 需要使用 `\\`

另外，如果你使用 `printf` 输出的话，额外有一个转换：

输出 `%` 需要使用 `%%`

输入

無

输出

```
~~~\"'(^.^ )'"/~~~ByeBye!!
```

下面这个字符画不是输出哦

纯狐也向你挥了挥手哦！

```
.....:~\"i1111Y***1111111\"\\,,:.....
.....
.....:~\"i1*YF%FX*11i>~'~>>>>i1*YYYXFY*1\":::.....
.....
.....:~i1FFXY*1i\"~\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'~~~~~\\~~'~!ii*F%%*1\"\\:~...
.....
.....:~11YFY1>'\\\\\\\\\\\\'>>11111111]]]]]]*1YYYYYYYYY*i\\\\>]]]]1*XF1i`
.....
.....:~1XY1i~\\\\\\\\\\\\~1*YYYYYYY1i~\\\\'!i]]]]*YYI1!ii!\\\\~]]]]>~iXF
1.....
.....~1%Y1~\\\\\\\\\\\\\\\\!YYYYYYYIi\\\\!*YYYI]]]]]]YYIi\\\\\\\\\\\\'1*]]]]1\\\\x%
%F\\.....
.....`1F%1'\\\\>>~\\\\\\\\\\\\!YYYYYYYi\\\\!YYYYY]]]]]]*YYY1ii11Y1]]]]1>~1F1
`1%1\"11\":.....
.....`1%i\\\\'>]>\\\\\\\\\\\\i*YYYYYYY1'\\\\1YYYY1]]]]]]*YYYYYYY*~i>]]]]>~>X1\\
.,1%~%~%F1\\.....
```

[illegible]

Author:星辰的微光

题目描述

假设从外面进入学校和从校内出来到校外都需要花费一个单位的时间，求所有人都进入学校最少需要花费几个单位的时间。

输入描述:

输出描述:

测试样例

输入

```
3
6 3
10 10
10 1
```

输出

```
5
1
-1
```

author: HiDen

C 唐朝诡事录之逃出密室

题目背景

小羊学姐又又又刷完一遍电视剧《唐朝诡事录》之后，她心满意足地上床睡觉了。这一次她梦到自己身处一个密室，旁边站着狄公弟子苏无名和中郎将卢凌风，一番询问之后，了解到：在探查案件期间，他们被困入了一个密室，需要解开一个题目才能逃出去。虽然小羊学姐知道自己的探查能力远远不如两位先辈，但是她还是想发挥自己学习信息类专业的优势，利用计算机来帮助解决题目，一起逃出密室。

题目描述

密室墙壁上刻有一个数字（十进制），旁边有一串操作指示，需要按照操作指示处理这个数字，处理后得到的数字（十进制）就是出口大门的密码。

操作指示：首先，将原数字（十进制）转为二进制数，最低位编号为 0。接着，一共有若干条指令，每一条指令含有 3 个数字： op ， $high$ ， low 。该指令含义是将上一步得到的二进制数的编号 $high$ 至 low 位进行 op 变换（ op 变换的具体含义见如下表格）。

op值	对应变换
0	置0
1	置1
2	1变为0,0变为1

输入

输入第一行，一个十进制正整数（无前置0），在 `int` 范围之内；

接下来有若干行（不超过100行）输入，每行包括三个非负整数，分别表示一条指令的 op ， $high$ ， low （范围 $0 \leq low \leq high \leq 30$ ）。

输出

输出一行，一个十进制整数（无前置0），表示密码。

输入样例

```
8
0 3 0
1 2 1
2 4 0
```

输出样例

```
25
```

Hint

将十进制数 num 的二进制表达形式（最低位编号为 0）的第 i 位置0: `num & (~ (1 << i))`;

将十进制数 num 的二进制表达形式（最低位编号为 0）的第 i 位置1: `num | (1 << i)`;

将十进制数 num 的二进制表达形式（最低位编号为 0）的第 i 位翻转: `num ^ (1 << i)`;

D ljh的成绩分析

题目描述

老家伙ljh又在分析同学们的上机成绩了，现在ljh得到了班上 n 名同学的成绩序列，已知本次上机满分为 110 分，ljh想将成绩序列**从小到大排序**，计算所有同学的**平均分**，并统计**有多少名同学AK(取得满分)**。

输入

第一行，一个整数 n ，意义如题目所述；

第二行， n 个整数，即所给成绩数列 a_i ，相邻两个数以空格分隔。

输出

第一行， n 个整数，即从小到大排序后所得的成绩序列；

第二行，1 个**浮点数**，保留至**小数点后两位**，即计算所得的平均分；

第三行，如果没有同学AK，则输出 `It's a great pity that no one got a perfect score.`，否则输出 `There are x people who got full marks!`，其中 x 为一个整数，即AK人数；

输入样例1

```
6
110 85 60 95 104 40
```

输出样例1

```
40 60 85 95 104 110
82.33
There are 1 people who got full marks!
```

输入样例2

```
2
80 90
```

输出样例2

```
80 90
85.00
It's a great pity that no one got a perfect score.
```

数据范围

$1 \leq n \leq 5000, 0 \leq a_i \leq 110$ 。

AUTHOR: ljh

E void排昵称

题目背景

void在网络上有许多的昵称，某天他突然想起来，想要把这些昵称按照字典序升序排一遍，于是他找到了你。

- 首先，我们定义什么叫做字典序
- 两个字符串 $str1, str2$ 的字典序大小，一般情况下等价于二者**第一个不相同的字符的大小**；特别地，若 $len(str1) > len(str2)$ ，且 $str1$ 的前 $len(str2)$ 个字符恰好为字符串 $str2$ ，则 $str1$ 的字典序大于 $str2$ 。如：`a < b`，`az < b`，`abcz < abd`，`abc < abca`，`abac = abac`。
- 或者说，我们将两个字符串中较短的那个在末尾补上0使其与较长的长度相等，然后从前往后比较，如果有某一位上 $str1$ 的字符大于 $str2$ 的字符，那么 $str1$ 的字典序也就大于 $str2$ 。

没有思路可以去仔细看Hint

输入

输入若干行，每行一个字符串，仅由小写字母组成

输出

输出若干行，每行一个字符串，为输入的所有字符串按照字典序升序排名之后的结果

样例

样例输入

```
avoid
evoid
cvoid
bvoid
dvoid
```

样例输出

```
avoid
bvoid
cvoid
dvoid
evoid
```

数据范围

$1 \leq \text{输入字符串数} \leq 100$

$1 \leq \text{字符串长度} \leq 100$

Hint

可以用二维数组存储若干个字符串，每行存储一个字符串。

比较两个字符串的字典序可以用 `string.h` 中的库函数 `strcmp`：设 `str1`, `str2` 为两个字符串，`strcmp(str1, str2)` 返回值 (`int` 型整数)：

- 正数：表示字典序 $str1 > str2$ ；
- 负数：表示字典序 $str1 < str2$ ；
- 0：表示字典序 $str1 = str2$ 。

思路1-快速排序：

推荐大家按照此思路完成本题。

可以用库函数 `qsort` 对二维数组进行排序，其中：

- 第一个参数应该是用来存储若干字符串的二维数组的数组名；
- 第二个参数应该是字符串的数量；
- 第三个参数应该是二维数组每行的占用的字节数，可以用 `sizeof(数组名[0])` 表示；
- 第四个参数 `comp` 函数可以直接使用库函数 `strcmp`，也可以自己写，如下：

```
int comp(const void *p1, const void *p2) {
    char *a = (char *)p1; //将p1强制转换为char*类型的指针
    char *b = (char *)p2; //将p2强制转换为char*类型的指针
    return strcmp(a, b); //返回两个字符串的字典序大小
}
```

思路2-冒泡排序

冒泡排序对 `int` 型一维数组进行排序的函数模板：

```
void bubbleSort(int a[], int n) {
    int i, j, hold;
    for(i = 0; i < n-1; i++)
        for(j = 0; j < n-1-i; j++)
            if(a[j] > a[j+1]) {
                hold = a[j];
                a[j] = a[j+1];
                a[j+1] = hold;
            }
}
```

改为对存储字符串的二维数组排序，需要的改动如下：

- 传入数组参数应为 `char a[][每行长度]`；
 - 第二个参数应该是字符串的数量；
 - 临时变量 `hold` 需要声明为 `char` 型数组，即 `char hold[数组长度]`；
 - 判断 `a[j]` 和 `a[j+1]` 的字典序大小可以用 `strcmp` 函数，即将 `if` 的条件改为 `strcmp(a[j], a[j+1]) > 0`
 - 交换 `a[j]` 和 `a[j+1]` 可以用 `strcpy` 函数，将赋值语句 `x = y` 改为 `strcpy(x, y)`
- 可见还是 `qsort` 函数好用呀！

F 幸运日

题目描述

某只buaaer仍沉浸在buaa70周年校庆中，尽管10月25日参加了一天校庆活动，但仍意犹未尽。于是她决定把所有包含子串“102”的日期作为她的幸运日。

规定使用形如 `YYYYMMDD` 的八位数字表示一个日期，前4位代表年，接下来2位代表月，最后2位代表日。当这个8位数包含“102”时即为幸运日，如 20221025。

现在，给定两个日期 $D1, D2$ ($D1 \leq D2$ 且 $|D2 - D1| \leq 100000$ 且 $10000101 \leq D1 \leq D2 \leq 99991231$)，请你帮忙计算在 $D1$ 和 $D2$ 之间有多少个幸运日（包含 $D1$ 和 $D2$ ）。

输入格式

本题存在多组数据（数据组数 ≤ 10 ）。

对于每组数据：

一行，两个8位整数 $D1, D2$ ，中间用空格隔开， $D1 \leq D2$ 且 $|D2 - D1| \leq 100000$ 且 $10000101 \leq D1 \leq D2 \leq 99991231$ 。

输出格式

对于每组数据分别输出答案：

一行，一个整数表示在 $D1$ 和 $D2$ 之间的幸运日的天数（包含 $D1$ 和 $D2$ ）。

样例输入


```
20220101 20221225
19370207 20131028
20931022 20950228
```

样例输出

```
12
1145
22
```

样例解释

对于第一组数据 20220101 20221225 符合条件的幸运日有 20220102, 20221020~20221029, 20221102。

Hint

- 本题数据保证暴力能过。
- 对于日期 *date*，如何判断其是否含有 102：可以将 *date* 转换为字符串，当 `strstr(s, "102") != NULL` 时说明存在子串 102。另外，将 `int` 型转换为 `char` 型有库函数 `atoi(str)`，但oj并不支持。在这里我们提供另一种便捷的做法：`sprintf(str, "%d", num)` //将num转为字符串输入到str中

即

```
int islucky(int date) {
    char r[10];
    sprintf(r, "%d", date);
    return strstr(r, "102") == NULL ? 0 : 1;
}
```

G 保护大结界

题目描述

Hakurei 大结界遭到了破坏，一些外界的生物入侵了幻想乡，而 Hakurei Reimu 正忙着[收集信仰](#)，毫不知情。无奈之下，她的好友 Kirsame Marisa 带着河童的新式装备来到了大结界边境来对抗入侵者。

Kawashiro Nitori 对 Marisa 八卦炉的导弹进行了改良，可以根据发射时敌人的坐标进行追踪和精确打击。由于大结界对幻想乡的封闭性，边境处的空间是扭曲的，距离的计算不能被常识所束缚。为了方便随时获取战场的信息，Nitori 给 Marisa 的终端上传了距离计算公式：

$$d((x_1, y_1, z_1), (x_2, y_2, z_2)) = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$$

为了节省八卦炉的能量，Marisa 可以在发射导弹之前飞到某个位置用体术消灭此处的所有敌人。不过因为时间紧迫，她只能这样消灭敌人一次，剩下的敌人需要用导弹一次性歼灭；当然，她也可以选择直接发射导弹。她想知道，自己应该在何处时，与剩余每个敌人的距离之和最小，并计算出这个最小值以便充能。

输入

第一行，一个正整数 n ，表示敌人的数量。

接下来 n 行，每行三个整数 x_i, y_i, z_i ，表示第 i 个敌人的坐标。

输出

一行，一个整数，表示在扭曲空间中 Marisa 与所有敌人距离之和的最小值。

样例

输入

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

输出

```
6
```

数据范围

$1 \leq n \leq 10^5, |x_i|, |y_i|, |z_i| \leq 10^5$ 。

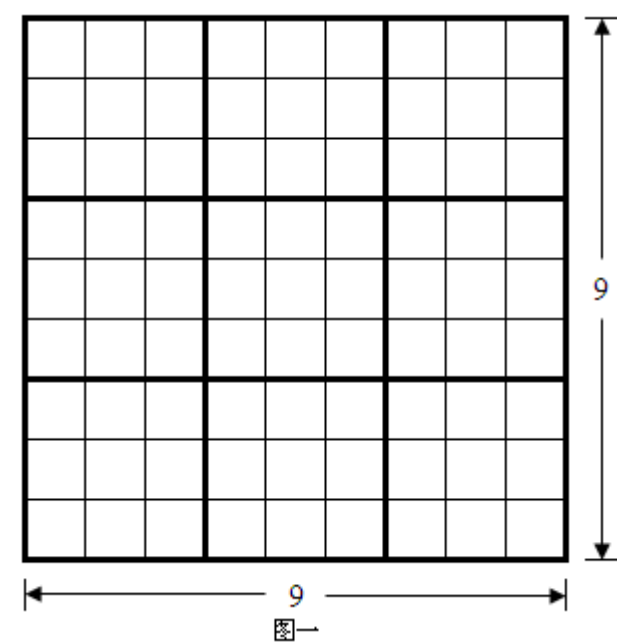
NOT HINT

扭曲空间中的距离看起来比实际距离简单了不少。

H Red的数独谜题

题目描述

数独，又称“九宫格”，是一种运用纸、笔进行演算的逻辑游戏。游戏目标是将9×9盘面上的数字填满，每一行、每一列、每一个粗线宫内的数字均含1-9，不重复。



Red 在纸上填好了一个正确的数独，但他悄悄替换了其中两个数字的位置。他想知道替换后的数独是否正确，你能帮帮他吗？若不正确，请你找到他替换的两个数字的位置。

输入格式

9 行，每行 9 个数字，表示数独盘面。

输出格式

一行。若给出的数独符合规则，则输出 `Congratulations!`，否则输出形如 `(x1,y1) (x2,y2)` 的字符串，表示第 x_1 行 y_1 列的数字与第 x_2 行 y_2 列的数字交换后，得到的数独符合规则，其中， (x_1,y_1) 与 (x_2,y_2) 按字典序排列。

样例输入

```
2 1 3 4 5 6 7 8 9
4 5 6 7 8 9 1 2 3
7 8 9 1 2 3 4 5 6
2 3 4 5 6 7 8 9 1
5 6 7 8 9 1 2 3 4
8 9 1 2 3 4 5 6 7
3 4 5 6 7 8 9 1 2
6 7 8 9 1 2 3 4 5
9 1 2 3 4 5 6 7 8
```

样例输出

```
(1,1) (1,2)
```

Aurthor : Red

I 求求你了帮帮可莉吧1

题目描述

可莉作为全提瓦特最能被关禁闭的孩子，只有靠做数学题打发时间，现在可莉想算两个大整数相乘的结果，但是可莉的手算能力比较差，你能帮帮可莉吗？

输入描述

一共两行，每行一个大整数。每个整数的字符串长度 $len, 1 \leq len \leq 1000$ 。保证整数不含前导零，且每个整数都是非负整数。

输出描述

一个大整数，为两个整数相乘的结果。输出不要含有前导零。

输入样例

```
114514
415411
```

输出样例

```
47570375254
```

Hint

将整数当做字符串读入。



J 指尖宇宙

题目背景

小小的指尖中，蕴藏着大大的宇宙。最开始，宇宙是一个混沌的奇点，但当你用指尖去衡量她时，她可以被放大，可以被缩小，亦可以被替换。

题目描述

宇宙可以看作一个字符矩阵（仅有可见字符与空格构成）。初始状态下，宇宙有 m 行 n 列。 $Jery$ 将对宇宙进行 T 次操作，他想知道这 T 次操作之后的宇宙长什么样子。

具体来说，操作有以下几种：

指尖膨胀

指定一个**可见字符**，将宇宙中所有该字符替换成一个 m_0 行 n_0 列的矩形，同时，其它位置都应该等比放大，且用该位置原来的字符填充该矩形。

注意，进行该操作后，矩阵的行数会变为原来的 m_0 倍，列数会变成原来的 n_0 倍。（即使宇宙中不存在指定的字符，宇宙每个位置也会按该比例放大）

举个例子，假如原来的宇宙为一个 3 行 5 列的字符矩阵：

```
## 01
#0.&&
0  #
```

指定字符 `#` 并将其替换为一个 2 行 4 列的字符矩阵：

```
aA B
C K0
```

替换后的结果为一个 6 行 20 列的字符矩阵：

```
aA BaA B      00001111
C K0C K0      00001111
aA B0000....&&&&&&&&
C K00000....&&&&&&&&
0000              aA B
0000              C K0
```

指尖湮灭

指定一行（列），将该行（列）消除，上下（左右）合并。

举个例子，假如原来的宇宙为一个 3 行 5 列的字符矩阵：

```
## 01
#0.&&
0  #
```

指定第 4 列消除，则结果为一个 3 行 4 列的字符矩阵：

```
## 1
#0.&
0  #
```

指尖坍缩

指定一个 r 行 c 列的字符矩阵作为坍缩因子，同时指定一个字符 *ord* 作为秩序符号，以及一个字符 *mes* 作为混沌符号。设宇宙为 M 行 N 列，这里保证 r 是 M 的因子， c 是 N 的因子。

将宇宙按 r 行 c 列为一个单位分成若干块，坍缩后，每一个单位块均会变成一个字符，于是宇宙变成 $\frac{M}{r}$ 行 $\frac{N}{c}$ 列。

对于每一个单位块，如果它与坍缩因子相同（这里的相同指这 $r \times c$ 个字符逐位置比较均相同），那么将它变成秩序符号 *ord*。否则，判断该单位块中非空格的字符类型数 d ：

- 如果 $d = 0$ ，即该单位块全都是空格，那么将该单位块变为一个空格；
- 如果 $d = 1$ ，即除空格之外有且仅有一种字符，那么将该单位块变为该字符；
- 如果 $d \geq 2$ ，即除空格之外有至少 2 种字符，那么将该单位块变为混沌符号 *mes*。

举个例子，假如原来的宇宙为一个 2 行 6 列的字符矩阵：

```
##0  #
#00. #
```

指定字符 **P** 和 **@** 分别作为秩序符号 *ord* 和混沌符号 *mes*。并指定坍缩因子为一个 2 行 1 列的字符矩阵：

```
#
#
```

替换后的结果为一个 1 行 6 列的字符矩阵：

```
P@0. P
```

输入

第一行 3 个整数： m , n , T 。

接下来 m 行，每行 n 个字符，表示原始宇宙。

然后输入 T 组操作：

- 对于指尖膨胀
 - 第一行 2 个字符，分别为 **1** 和指定的字符；
 - 第二行 2 个整数，分别为 m_0 和 n_0 ；
 - 接下来 m_0 行，每行 n_0 个字符，表示替换的矩阵；
- 对于指尖湮灭
 - 第一行 2 个字符，**2c** 或 **2r**，其中，**2c** 表示删除列，**2r** 表示删除行；
 - 第二行 1 个整数，表示删除的行数（列数）；
- 对于指尖坍缩
 - 第一行 3 个字符，分别为 **3**，秩序符号 *ord* 和混沌符号 *mes*；
 - 第二行 2 个整数，分别为 r 和 c ；
 - 接下来 r 行，每行 c 个字符，表示坍缩因子；

输出

T 次操作之后的宇宙。

样例

输入 #1

```
2 2 2
@#
k
1@
2 3
k #
###
3 ?
2 2
##
##
```

输出 #1

```
?
kk
```

样例解释

初始宇宙：

```
@#
k
```

每次操作后宇宙变化如下：

操作 1：指尖膨胀

```
k #####
#####
    kkk
    kkk
```

操作 2：指尖坍缩

```
?
kk
```

数据范围

操作过程中，保证宇宙面积不超过 10^6 ，且操作次数不超过 10。