

C8 - Statement

A 诚信守则

诚信承诺

我承诺：诚信考试，遵守考试规定，自觉维护考场秩序，包括但不限于：

1. 考试全程不使用个人电话，不做与考试无关的事情。
2. 个人独立答题，不通过网络获取答案或提示；除监考老师以外，不与他人进行任何形式的交流，不使用任何实时通信软件或网站（包括但不限于QQ、微信、在线共享文档）。
3. 考试时按监考老师指示进行考试系统的登录，不尝试任何异常登录行为（尝试登录别人的帐号，个人账号在多个设备登录），妥善保管好个人的帐号和密码并保证不透露给他人。

阅读完“诚信承诺”后，请按照下述要求完成本题

输入

一行一个字符串，保证字符串为 `If one loses honesty, he will lose everything.` 或者 `I promise I do not cheat in the exam.`。

输出

输出一行。

如果输入为 `If one loses honesty, he will lose everything.`，则输出 `I promise I do not cheat in the exam.`。

如果输入为 `I promise I do not cheat in the exam.`，则输出 `If one loses honesty, he will lose everything.`。

输入样例 1

```
If one loses honesty, he will lose everything.
```

输出样例 1

```
I promise I do not cheat in the exam.
```

输入样例 2

```
I promise I do not cheat in the exam.
```

输出样例 2

```
If one loses honesty, he will lose everything.
```

B Softplus

Hint

注意 `"`、`'` 和 `\` 要用转义字符输出。

Not Hint

“Hello, World”，从最初那一行简单的代码开始，你已经迈出了探索编程世界的重要一步，这段旅程充满了成长和惊喜。

愿你在这条编程之路上始终保持对技术的热爱与好奇，代码如诗，创意无限。

愿你带着最初那一行简单的代码带给你的欣喜，不断前行。

Author：Moca

D 位运算计算器

题目背景

请你实现一个位运算计算器，能够计算 32 位无符号数按位与 `&`、按位或 `|` 和按位异或 `^` 三种位运算操作，并统计最终结果的 32 位无符号数补码表示中 1 的个数。

输入格式

不定组数据输入，保证数据组数不超过 10000。

每组数据一行，先输入一个 `unsigned int` a ，再输入一个字符 op ，最后再输入一个 `unsigned int` b 。

op 一定为 `&`、`|` 或者 `^` 中的一个，且前后均有一个空格。

输出格式

对于每组数据，输出一行两个整数，分别表示对应的运算结果，和运算结果的 32 位无符号数补码表示中 1 的个数。

输入样例

```
1 & 2
1 | 2
1 ^ 2
```

输出样例

```
0 0
3 2
3 2
```

Hint

可以参考如下代码，获取一个 `unsigned int` 的每一位

```
unsigned int n = 14;
for(int i = 31 ; i >= 0 ;i--) {
    printf("%u", (n >> i) & 1);
}
```

E 小懒獭与半素数

问题描述

小懒獭最近发现了一类特别的数字，叫做半素数。

一个正整数被称为**半素数**，如果它是两个（不一定不同）素数的乘积。例如： $10 = 2 \times 5$ 、 $9 = 3 \times 3$ 是半素数，而 $8 = 2 \times 2 \times 2$ 、 $105 = 3 \times 5 \times 7$ 不是半素数。

请你帮小懒獭实现一个程序，用来判断一个数是不是半素数。

输入格式

不定组输入，保证组数不超过 1000。

对于每组输入，输入一行一个正整数 n ，保证 $1 \leq n \leq 10^8$ 。

输出格式

对于每组输入，输出一行，如果 n 是半素数，由小到大输出 n 的两个素数因子；如果 n 不是半素数，输出 `Not otter's type!`。

输入样例

```
10
9
8
105
```

输出样例

```
2 5
3 3
Not otter's type!
Not otter's type!
```

Hint

验证一个数 n 是不是质数，可以通过只验证在 \sqrt{n} 范围内有没有因数就可以哦。同样地，如果一个数字是半素数，那么它一定有一个素因子不大于 \sqrt{n}

F 扫雷 2024

问题描述

扫雷是一个经典的游戏。游戏的规则是，玩家点击一个格子后，如果该格子下面没有地雷，格子会显示一个数字，表示其周围 8 个方向内有多少颗雷；如果该格子下有地雷，则游戏结束。

你需要模拟一个扫雷的场景，给定地图的大小和所有雷的位置，输出最后生成的扫雷地图。

输入格式

第一行输入两个整数 n 和 m ，表示扫雷游戏的矩阵的行数和列数， $1 \leq n, m \leq 50$ 。

第二行输入一个正整数 k ，表示有多少个雷，保证 $1 \leq k \leq n \times m$ 。

接下来 k 行，每行两个正整数，表示每个雷的位置 (x, y) ，保证 $1 \leq x \leq n, 1 \leq y \leq m$ 。保证输入的雷互不重复。

输出格式

输出 n 行，每行 m 个字符，若字符对应的位置是雷，则输出 `*`，否则输出一个整数表示周围八个方向内有多少颗雷。

输入样例

```
3 3
4
1 1
1 3
3 1
3 3
```

输出样例

```
*2*
242
*2*
```

Author: Moca

G 排行榜整理

题目描述

信息大类的程序设计课也迎来了尾声，OJ 管理员想通过同学们在上面的通过题目数量和总罚时时间整理排名，请 OJ 排行榜名列前茅的你来编写代码帮助他。

排名的规则是通过题目数量越多，排名越高，若题目数量相同，则罚时少者排名靠前，若二者全部相同，则排名相同（注意：假如两名同学并列第一，那么排名仅次于他们的同学排名为第三名，以此类推）。

输入

第一个数为同学的总数

接下来 n 行，每行 2 个整数 a, b ，其中 a 为该位同学通过题目数量， b 为该位同学罚时时长。

输出

一共 n 行，每行第一个整数为排名，后接一个冒号和空格，紧接着用空格分开的通过题目数和罚时，详见下面样例

输入样例1

```
5
10 6
9 8
9 4
20 10
6 0
```

输出样例1

```
1: 20 10
2: 10 6
3: 9 4
4: 9 8
5: 6 0
```

输入样例2

```
5
10 6
9 8
9 8
20 10
6 0
```

输出样例2

```
1: 20 10
2: 10 6
3: 9 8
3: 9 8
5: 6 0
```

数据范围

对于30%的数据， $n \leq 1000$

对于全部的数据， $n \leq 500000$, $0 \leq a \leq 10000$, $0 \leq b \leq 1000000$

hint

涉及两个关键字的排序，qsort 的 cmp 函数可以参考下面来进行书写：

```
int cmp(const void *p1,const void *p2){
    int *a1=(int*) p1;
    int *a2=(int*) p2;
    if(a1[0]==a2[0]){
        return a1[1]>a2[1]?1:-1;    //第二关键字的比较
    }
    return a1[0]<a2[0]?1:-1;    //第一关键字的比较
}
```

Author: zsm666

H Orch1d 的最优序列

题目背景

“结束了吗？”

“结束了。”

“真的结束了吗？”

“好像还没有。”

Orch1d 收到了两年前的自己发来的时空邮件，打开 *accoding*，看着已结束的 2022 级 *C* 语言程序设计第一次上机赛，一切仿佛发生在昨天。

题目描述

邮件里是 Orch1d 留下的一句话，由若干**英文单词**组成，每个单词间用**空格**隔开，保证单词**仅由大小写字母组成**，现在我们可以将所有单词按一定顺序排列并连接。

例如：*With An Orchid* 可以有 *WithAnOrchid*, *AnOrchidWith*, *OrchidWithAn* 等排列方式，要求每个单词**必须**且**仅能**出现一次，请输出所有排列方式中字典序最大的结果。

输入

一行由若干英文单词组成的句子

输出

一个仅由大、小写字母组成的字符串，表示字典序最大的排列结果。

输入样例1

```
b bu buaa
```

输出样例1

```
bubuaab
```

输入样例2

```
practice makes perfect
```

输出样例2

```
practiceperfectmakes
```

输入样例3

```
Never Fades love
```

输出样例3

loveNeverFades

数据范围

保证输入的单词数不超过 9，每个单词的长度不超过 9。

Author：With An Orch1d

I 小水獭与大作业

注意本题的内存限制

题目背景

要到学期末了，小水獭们也不得不开始做他们的大作业了。

经过讨论，小水獭们决定做一个共享文档网站，以实现知识的共享。在这个网站中，每个用户可以通过输入一个房间号来进入对应的文档房间。如果用户输入的房间号之前没有出现过，那么系统会创建对应的房间，初始内容为空。用户进入房间后，可以在已有的字符串后面继续追加新字符串，共同完善这个共享文档。

请你帮助小水獭们完成这个网站。

输入格式

第一行一个正整数 t ，表示接下来将要进行的操作数，保证 $1 \leq t \leq 1000$ 。

接下来输入 t 次操作，每次操作由两行字符串组成，第一行表示房间号 $room$ ，保证 $room$ 的长度在 $[1, 10]$ 范围内，且仅由 ASCII 在 $[33, 126]$ 的字符组成。第二行表示追加的字符串 S ，保证 S 的长度在 $[1, 10000]$ 范围内， S 仅由 ASCII 在 $[32, 126]$ 的字符组成，但保证开头和结尾一定不是空格。

保证所有的 S 加在一起不超过 400000，每个房间最后的内容不超过 10000。

输出格式

所有操作完成后，按照房间出现的顺序，对所有房间，输出如下格式的两行内容：

房间名：
房间内的内容

输入样例

```
5
accoding
If one loses honesty, he will lose everything.
moca
Little Lazy Otter al
accoding
I promise I do not cheat in the exam.
Math
1 + 1 = 2.
moca
so study hard!
```


输出样例

```
accoding:
If one loses honesty, he will lose everything.I promise I do not cheat in the exam.
moca:
Little Lazy Otter also study hard!
Math:
1 + 1 = 2.
```

样例解释

- 第一次操作，输入房间号 `accoding`，还不存在对应的房间，所以创建房间 `accoding`，房间的内容变为：`If one loses honesty, he will lose everything.`
- 第二次操作，输入房间号 `moca`，还不存在对应的房间，所以创建房间 `moca`，房间的内容变为：`Little Lazy Otter al`
- 第三次操作，输入房间号 `accoding`，存在对应的房间，在后面追加内容，房间的内容变为：`If one loses honesty, he will lose everything.I promise I do not cheat in the exam.`
- 第四次操作，输入房间号 `Math`，还不存在对应的房间，所以创建房间 `Math`，房间的内容变为：`1 + 1 = 2.`
- 第五次操作，输入房间号 `moca`，存在对应的房间，在后面追加内容，房间的内容变为：`Little Lazy Otter also study hard!`

操作结束，房间创建的顺序依次为 `accoding`、`moca` 和 `Math`，依次输出最后的内容

Hint

我们可以发现，根据总的房间数和房间内容的最大长度，理论上我们至少要开一个 1000×10000 的数组来存放所有的字符（这里没有算字符串最后的空字符），但是可以发现全部内容的总长度远远小于这个大小，如果真的这么做许多空间就被浪费了。

所以我们就想到，有没有什么办法能够动态地每次都开一个大小合适的字符数组来存放每个房间，我们就想到了课上讲到的动态存储分配：

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
char str[10];

int main()
{
    gets(str);
    int len = strlen(str);
    // 多开一位是为了存储字符串最后的空字符，直接写 len + 1 也可
    char *text = (char*)malloc(len*sizeof(char) + 1);
    strcpy(text, str);
    puts(text);
    free(text); // 有借有还

    return 0;
}
```

上面的代码展示了动态存储分配的一个基础使用方法，用来存储内容的 `text` 的大小就会根据内容的长度动态分配。在这道题中我们要存最多 1000 个房间的内容，所以我们可以定义一个指针数组 `char *otter[1001]` 来记录所有房间的内容，该数组每个元素 `otter[i]` 是一个 `char *` 类型的指针变量，当在某个房间后面追加内容时我们就用 `malloc` 申请合适大小的空间用来存放原来的内容加上新内容，并将原来申请的空间释放掉。

完成了大作业后，水獭们的程序设计之旅就暂时告一段落了。但小水獭们会在这里默默地为你们加油，期待看到你们用所学去探索更广阔的世界，在未来的旅途中书写更多的 "AC"。

会编程的人，运气不会太差。继续前进吧，未来属于敢于迎接挑战的你！



J 小僵尸吃脑子 2 （简单版）

题目描述

Paradise 很喜欢玩《僵尸大战植物》，这个游戏的目标是配合一只小僵尸吃掉脑子！最后一关中，Paradise 手滑将小僵尸传送到了一个庄园里……很不巧的是，这个庄园里除了脑子，还有若干个坚果墙和土豆雷

整个庄园可以看成一张 n 行 m 列的地图，左上角记为 $(1, 1)$ ，右下角记为 (n, m) 。地图的某处有一颗脑子，而小僵尸被传送到了 $(1, 1)$ 处。毫无疑问，它想要吃掉脑子！作为一只聪明的僵尸，它既可以消耗 1 秒移动到相邻的位置（不能跳出地图，也不能踩到土豆雷），也可以消耗 t 秒吃掉相邻的一个坚果。Paradise 想知道，它至少需要花费多少秒才能吃到脑子

输入

第一行，三个正整数 n, m, t
接下来 n 行，每行 m 个字符（不含换行），表示地图
地图中的字符有：

- `.` 表示一个空地，可以移动
- `#` 表示一个坚果，不能直接移动，但被啃掉后会变成空地
- `^` 表示一个土豆雷，不可以踩！
- `*` 表示脑子

输入保证 $(1, 1)$ 处一定是空地，整张地图有且只有一颗脑子，小僵尸一定能绕过所有土豆雷到达脑子的位置

输出

一个整数，表示小僵尸最少花费的时间

输入样例 1

```
5 9 4
...#^..^*
.#...#.^.
....#....
.##...##.
#.#.#.#.#
```

输出样例 1

```
14
```

输入样例 2

```
8 7 1
.#...#*
.##^##.
.#...#.
.#.#.#.
.#.#.#.
.#.#.#.
.#.#.#.
...#...
```

输出样例 2

```
8
```

输入样例 3

```
2 5 10
.###*
..^..
```

输出样例 3

```
34
```

样例解释

对于第一个样例，小僵尸可以选择 ~ 表示的路径移动

```
~~~#^,.^*
.#~.,#.^~
..~~#~~~~
.##~~~##.
#.#.#.#.#
```

对于第二个样例，小僵尸可以选择 ~ 表示的路径移动，途中吃掉了 2 个坚果。如果不吃坚果的话，它需要 30 秒才能到达脑子的位置，真是太慢了！

```
~~~~~*
.##^##.
.#...#.
.#.##.
.#.##.
.#.##.
.#.##.
.#.##.
...#...
```

对于第三个样例，小僵尸可以选择 ~ 表示的路径移动，途中吃掉了 3 个坚果。如果不吃坚果的话，就要踩到土豆雷了！聪明的僵尸永远不会踩雷！

```
~~~~*
..^..
```

数据规模

对于 30% 的数据， $1 \leq n, m \leq 10$
对于 100% 的数据， $1 \leq n, m \leq 100$, $1 \leq t \leq 10$

Not Hint

游戏可以通关，Paradise 也希望同学们的程序设计课程都能顺利过关！但程序设计永无止境，希望同学们在课程结束后还可以多加练习，进一步提高自己的编程水平！
下学期数据结构课程学过队列、广度优先搜索或堆等知识之后，欢迎同学们回来重新做一下这道题！对于每个测试点，你可以将你的程序优化到运行时间不超过 10 ms ！

题目描述

ddz 将 n 棵神奇的树自西向东间隔一米种成一排（最西边的树称作第 1 棵树），初始时每棵树高 0 米，第 i 棵树被浇一次水会长高 a_i 米，这 n 棵树的茂密程度 s 被定义为树高最低的那棵树的高度。

ddz 想偷懒，于是制作了一个机器人帮他浇水，初始时机器人在第 1 棵树的西边一米的位置，机器人会做以下两个操作中的一个：

- 往西移动一米并给当前位置的树浇一次水。
- 往东移动一米并给当前位置的树浇一次水。

但是机器人的电量并不是很多，只能执行 x 次操作，ddz 想知道在机器人执行完 x 次操作后，茂密程度 s 最大能达到多少。

补充说明：机器人可以移动到第 1 棵树的西边，也可以移动到第 n 棵树的东边。机器人只能做那两个操作中的一个，并不能做到直接在当前位置浇水等操作。

输入

第一个数为数据组数 t ($1 \leq t \leq 100$)

对于每组数据

第一行两个整数 n, x ($2 \leq n \leq 10^5, 0 \leq x \leq 10^{12}$)

第二行 n 个整数 a_i ($1 \leq a_i \leq 10^5$)

保证所有的 n 的和不超过 10^6

输出

对于每组数据，输出一行一个整数 s 表示茂密程度最大值。

输入样例

```
2
4 8
3 2 6 6
3 9
10 10 1
```

输出样例

```
6
4
```

Author: ddz